REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
06-07-2014	Final Report	1-Oct-2009 - 30-Mar-2014

4. TITLE AND SUBTITLE CMU-NREC Vehicle-Terrain Model Identification Program FInal Report	5a. CONTRACT NUMBER W911NF-09-1-0557
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 611102
6. AUTHORS Alonzo Kelly, Ammar Husain, Venkat Rajagopalan	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA 15213 -3589	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 55835-NS.6

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

This report outlines progress on program for the period Oct 1/2013 to Mar 30/2013.
It is also the final report for the program containing a program summary in the last section.

15. SUBJECT TERMS

system identification, vehicle autonomy, model predictive control, terrain classification, wheel-terrain interaction

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Alonzo Kelly
UU	UU	UU	UU		19b. TELEPHONE NUMBER 412-683-2550

Report Title

CMU-NREC Vehicle-Terrain Model Identification Program
FInal Report

ABSTRACT

This report outlines progress on program for the period Oct 1/2013 to Mar 30/2013.
It is also the final report for the program containing a program summary in the last section.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
08/31/2011	2.00	Neal Seegmiller, Forrest Rogers-Marcovitz, Greg Miller, Alonzo Kelly. A Unified Perturbative Dynamics Approach to Online Vehicle Model Identification, International Symposium on Robotics Research. 29-AUG-11, . : ,
08/31/2012	3.00	Neal Seegmiller,, Forrest Rogers-Marcovitz,, Alonzo Kelly. Online Calibration of Vehicle Powertrain and Pose EstimationParameters using Integrated Dynamics”, ICRA 2012. 14-MAY-12, . : ,
08/31/2012	4.00	Forrest Rogers-Marcovitz, Neal Seegmiller, Alonzo Kelly. Long?term Operation of Autonomous Robotic Systems in Changing Environments, Robotics: Science and Systems. 09-JUL-12, . : ,

TOTAL: **3**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received		Paper
10/05/2013	5.00	Neal Seegmiller, Forrest Rogers-Marcovitz, Greg Miller, Alonzo Kelly. Vehicle model identification by integrated prediction error minimization, International Journal of Robotics Research (04 2013)

TOTAL: **1**

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Alonzo Kelly	0.05	
FTE Equivalent:	**0.05**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME
Total Number:

Names of personnel receiving PHDs

NAME
Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
Venkat Rajagopalan	0.25
Ammar Hussein	0.25
FTE Equivalent:	**0.50**
Total Number:	**2**

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

see attachment

Technology Transfer

Real-Time Identification of Wheel Terrain Interaction Models for Enhanced Autonomous Vehicle Mobility

Final Progress Report, Apr. 24, 2014

National Robotics Engineering Center
Carnegie Mellon University

Alonzo Kelly, Ammar Husain, Venkat Rajagopalan

Foreword

This report outlines progress on program for the period Oct 1/2013 to Mar 30/2013.

It is also the final report for the program containing a program summary in the last section.

Table Of Contents

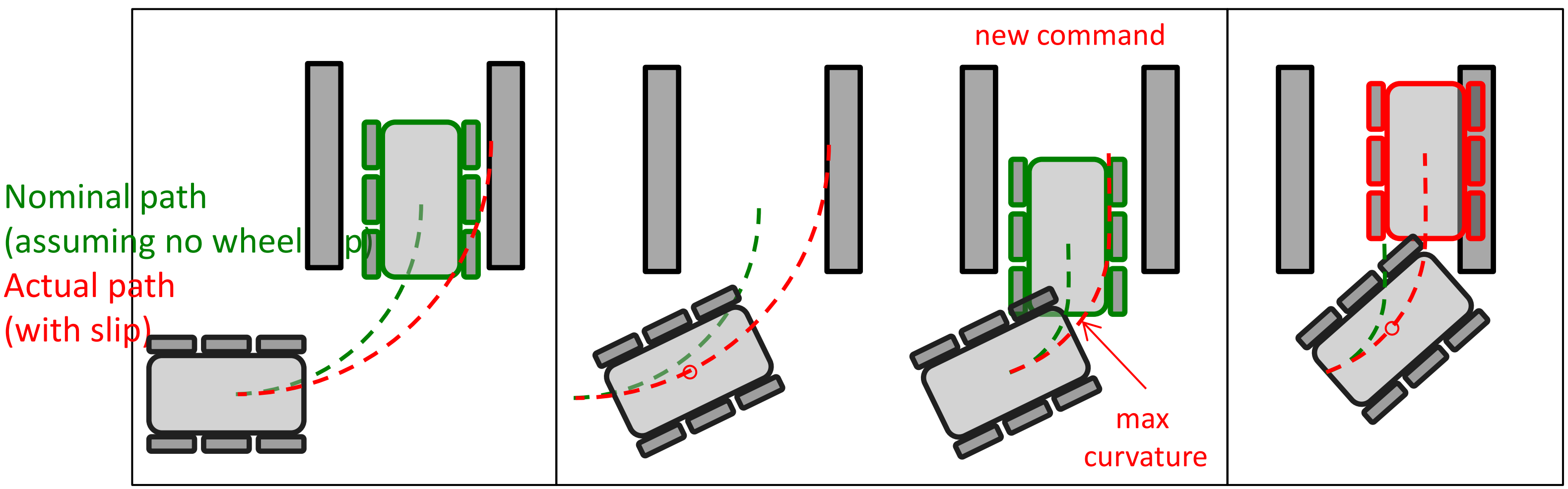
- Statement of Problem Studied
  - Motivation
- Summary of Most Important Results
- Metrics for Reporting Period
- Conclusion / Outlook

Statement of Problem Studied

- We address the problem of calibrating predictive models of ground vehicles in order to enable mobile robots (UGVs) to be more informed about their own mobility.
- UGVs predict their own motions as a basic aspect of every decision they make.
- Hence poor models present a fundamental barrier to high performance UGVs.

Motivation

- Here is a motivating example: a classic failure mode for the UPI program on the Crusher platform that motivated the original proposal.
- UGV is trying to make a very sharp turn but continually understeers. Feedback cannot remove the error because it is persistent. Eventually the turn becomes impossible. This occurred painfully often in some field tests. It would end a real UGV mission in failure.
- A system which can predict how much it understeers can simply compensate predictively and this problem is eliminated.
- The missing science is that slip depends on terrain mechanical properties (which depend on present and recent weather), vehicle, vehicle motion, slope, etc. and perception sensors cannot measure what is necessary. It has to be learned from experience. Hence our project.



Time t
Robot needs to turn sharply.

Time t+1
Robot understeers.
Now needs to turn more sharply.

Time t+2
Robot understeers again.
Now turn is impossible.

Table Of Contents

- Statement of Problem Studied
- Summary of Most Important Results
- Metrics for Reporting Period
- Conclusion / Outlook

Goals for Performance Period

- Our goals for this performance period were:
  1. improve our results for perceptual cueing and for slip model aided model predictive control.
  2. incorporate slip constraints into Differential Algebraic Equation (DAE) models of wheeled mobile robots.
  3. Publish Data Logs: Allow other researchers access to the multi-vehicle, multi-terrain data logs for research on vehicle mobility modeling.

Summary of Most Important Results

- The following large number of slides is organized analogously to those of the last report.
- However:
  - Results for this reporting period are far more definitive due to more extensive experiments using an excellent platform.
  - Most of the imagery is improved and related directly to the experiments performed.
- Part of the intent is for this material to perform double duty as the basis of subsequent publications.

Table Of Contents

- Statement of Problem Studied
- Summary of Most Important Results
  - Perception aided slip prediction
  - Model Predictive Control
- Metrics for Reporting Period
- Conclusion / Outlook

Table Of Contents

- Statement of Problem Studied
- Summary of Most Important Results
  - Perception aided slip prediction
  - Model Predictive Control
- Metrics for Reporting Period
- Conclusion / Outlook

Platform Details: Husky

- We used the Husky robot manufactured by Clearpath robotics to run the experiments on Perceptual Cueing as well as Model Predictive Control.
- Husky is a skid steered vehicle with the wheel rotation of both wheels on each side coupled together while remaining independent of the rotation of the 2 wheels on the other side. Skid steer was chosen because it is a high slip configuration relative to alternatives like Ackerman steer. Such platforms are more squarely Army relevant. Also, our work should matter more on such platforms.
- The platform has dimensions of 0.67m (Width) x 0.99m (Length) x 0.39m (Height) and drives at a maximum speed of 1m/s.
- The robot was retrofitted with a camera and a pose system along with an onboard computer.

Platform Retrofit

AVT GT1920C GigE Camera

Pose System: Novatel OEMV-3 GPS Receiver + Honeywell HG1930 IMU

Wheel Encoders

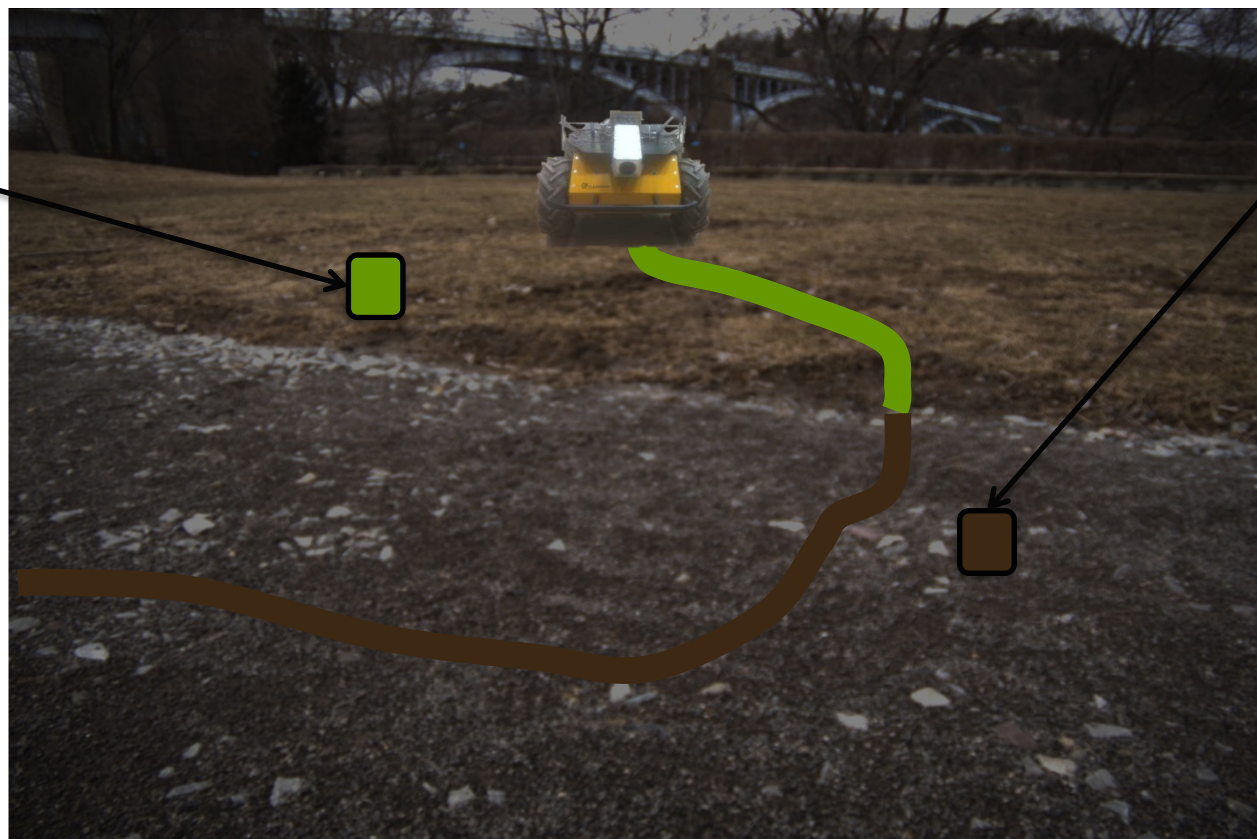
Perception aided slip Prediction

- Slip Prediction: We are trying to predict slip in pose estimation rather than slip in control. This approach allows us to eliminate dependence of our results on vehicle modeling error.
- The principle we employ is that the appearance and the slip characteristics of terrain are correlated, so experience with terrain in one visual class can be used to predict slip characteristics for upcoming terrain based on its appearance.
- In plainer terms once the system drives over gravel once, it should be able to switch on the gravel slip model if it can see that it is about to drive over gravel again.
- The important scientific questions are:
  - How correlated is the visual and mechanical signature.
  - (Related) how separable are the classes.
  - How much does adding vision improve slip prediction.
- In following slides, we evaluate the idea on a data set of color and slip data generated on our retrofitted Husky test vehicle.

Perceptual Cueing

- **We use the term to mean the use of perception to forecast and “cue up” the right models to switch on as terrain is traversed.**
- **Example:**
  - **Grass and Trail look visually different and they slip differently.**
  - **We can predict how a terrain would slip by visually classifying it.**

Grass:
Slips with Slip Function f_2()



Trail Road:
Slips with Slip Function f_1()

Unsupervised Approach

- It is important to note that we follow an entirely unsupervised approach for visually separating terrain types…
  - This ambitious approach does limit the learning ability of the algorithms relative to a human supervised approach.
  - However, it is significantly more practical because it can be done autonomously on highly diverse terrain.
- The alternative supervised approach is not practical in many cases. The process of labelling terrain types in numerous images would be too tedious in our application - especially on heterogeneous test sites.
- Another impact of this approach.
  - Lacking ground truth labels from a human supervisor, we have no basis for quantifying terrain classification error rates separately.
  - While this would be useful in a well engineered system, it is irrelevant in a proof of principle effort if the technique turns out to work anyway. Instead we concentrate on the bottom line - the slip position prediction error – and the technique does work anyway.
- If the system confuses gravel and asphalt but still predicts slip better, it is a success rather than a failure because the system can teach itself effectively without human intervention. An unsupervised success is far more significant to robotics and eventual deployment than a supervised one.

Table Of Contents

- Statement of Problem Studied
- Summary of Most Important Results
  - Perception aided slip prediction
    - Terrain Projection
    - Feature Generation
    - Classification
    - Class Selection
    - Learning Mobility Characteristics
    - Experimentation Setup
    - Experimental Results
  - Model Predictive Control
- Metrics for Reporting Period
- Conclusion / Outlook

Terrain Projections

- We create an elevation map of the terrain using elevation measurements from the onboard GPS
- Using the intrinsic & extrinsic camera to IMU calibration matrix, we transform a square grid in world coordinates (latitude, longitude, elevation) into a quadrilateral in image space (x_{pix}, y_{pix}), using matrix multiplication.
- Left: Geo-referenced elevation map, Right: Image from onboard camera

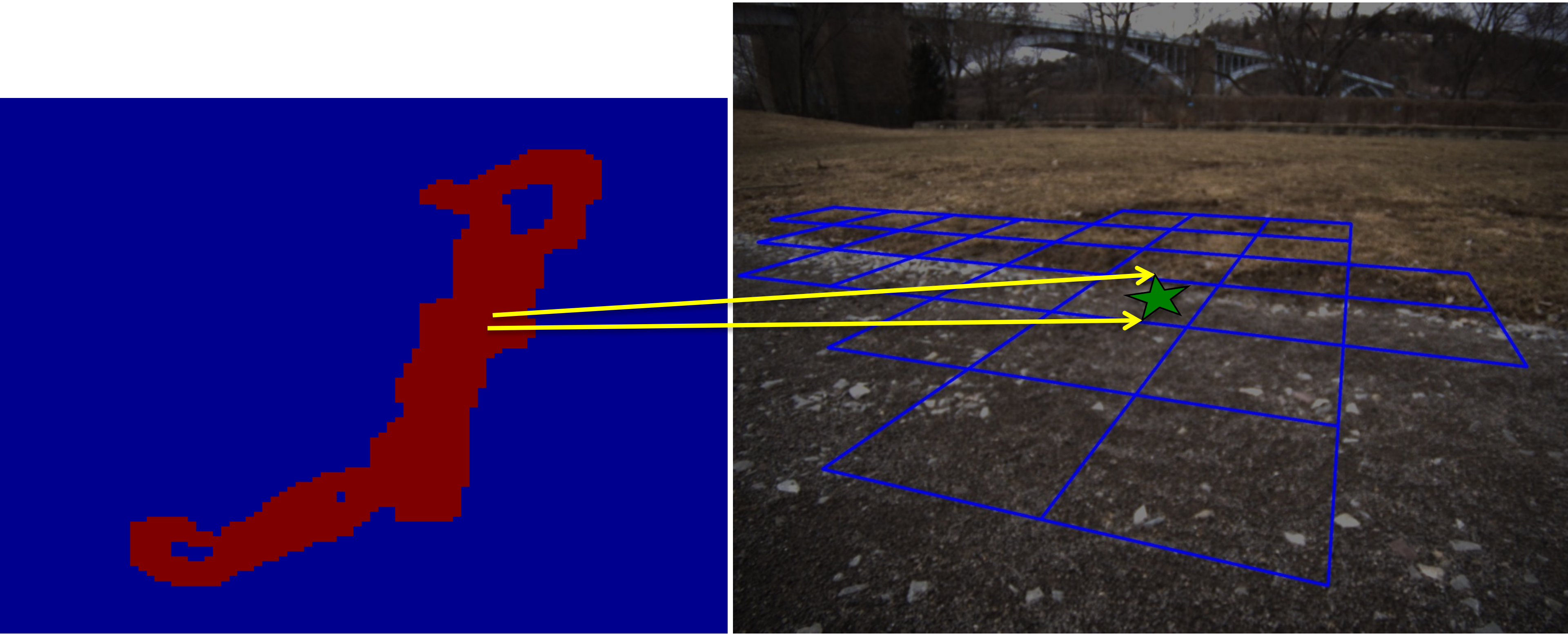


Table Of Contents

Feature Generation

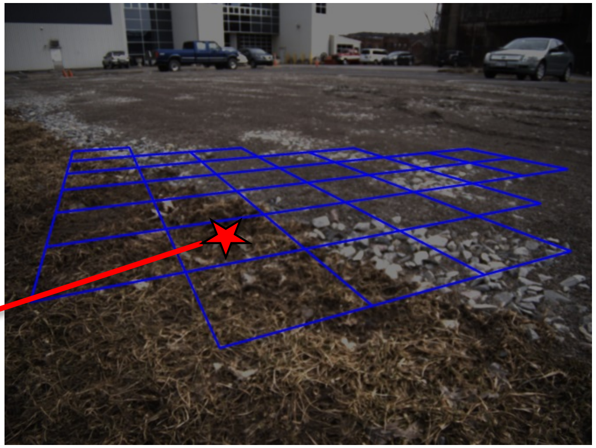
- Each quadrilateral of some fixed chosen size in an image is represented by a feature descriptor.
- This descriptor is simply a numeric vector that attempts to capture, in a few numbers, the visual information it contains.
- We implemented the following three feature generation algorithms, elaborated in subsequent slides. They are:
  I. F1: Color – L*, a* and b* channels
  II. F2: Color Textons
  III. F3: Maximum Response (MR8) Filter Bank Textons

Table Of Contents

Feature Generation F1: Color

F1: Color

For each patch calculate:
1. Mean (L, a, b)
2. Variance (L, a, b)
3. Sample size (w)



Feature Vector:

L mean	a mean	b mean

- **Color space is the space of red-green-blue points in 3D or some transformation of it.**
- **L,a,b is a color space designed to approximate human vision:**
- **L: represents the lightness of color,**
- **a: position between red/magenta and green**
- **b: position between yellow and blue**
- **The feature descriptor is computed by calculating the mean L, a and b values over all the pixels in the image quadrilateral.**

Table Of Contents

Feature Generation F2: Color Textons

Training Images

5 x 5 patch -> 3 channels = 75 element feature vector/ pixel

- **Textons refer to fundamental micro-structures in natural images and are considered to be the "atoms" of pre-attentive human visual perception.**
- **Several published studies exist on finding the best texton representation to describe texture in images.**
- **In this method, a texton is computed by obtaining a 5x5 local region in a 3 channel image, around each pixel in the image quadrilateral, producing a 75 element vector**

Texton Generation

- The "Bag of visual words" is a common technique in machine learning that creates a sparse vector of occurrence counts of a "vocabulary" of local image features.

- The vocabulary is generated on a set of training images by finding the exemplar descriptors over all training feature descriptors (75 element vector) using a famous algorithm called K-means.

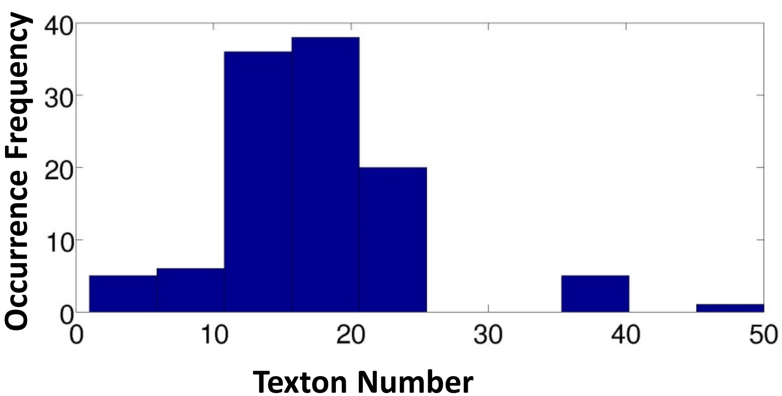
Using Color Textons

- This 75 element vector is mapped to its closest element (in Euclidean space) in a bag of visual words.
- For each image quadrilateral, a histogram of the frequency of matched texton vector is computed.
- With a Bag of words size of N, a histogram of N elements serves as the feature vector. We use 10 as the size of our dictionary.

For each pixel in patch calculate:

1. BOW sized vector (normalized)
2. Sample size (w)

Feature Vector:



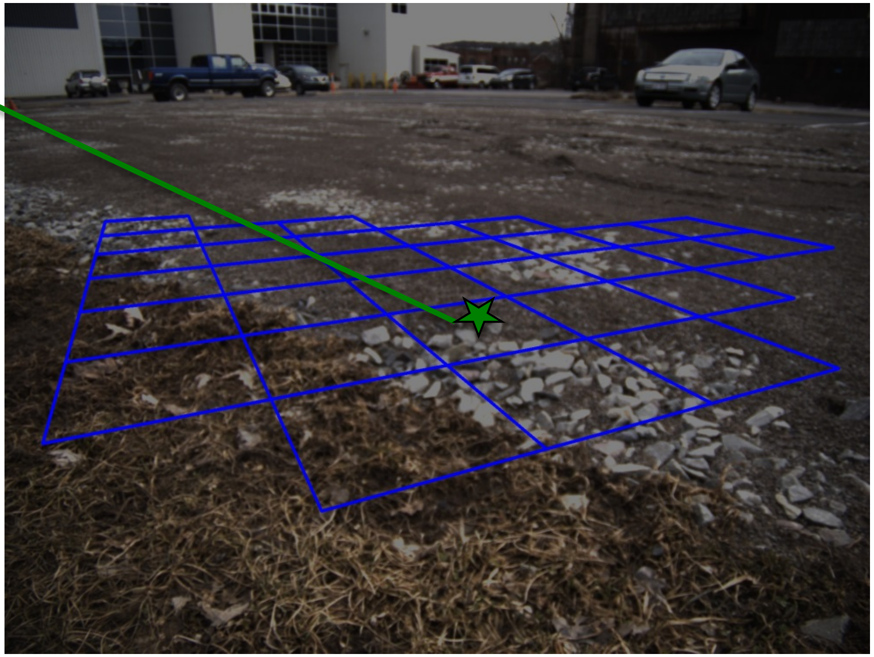
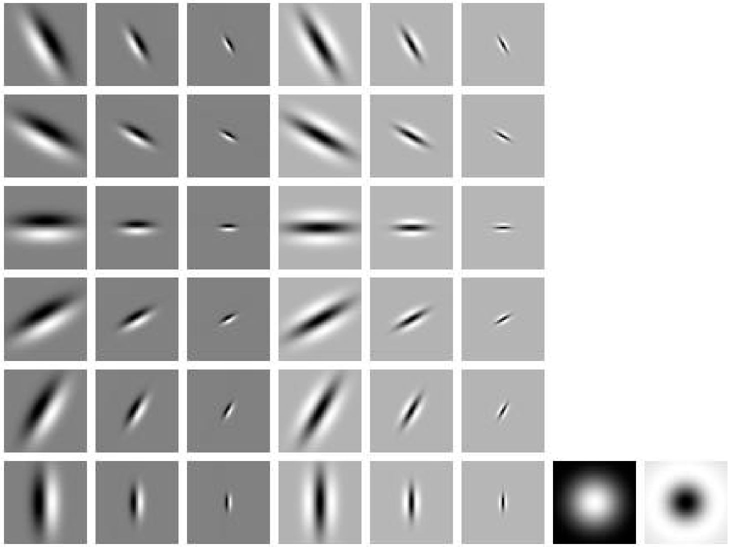


Table Of Contents

Feature Generation F3: MR8 Textons

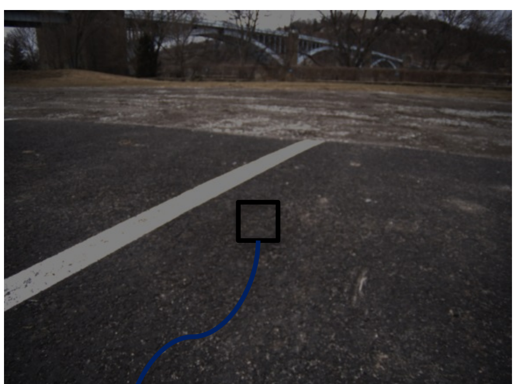
- MR8 Textons are computed by convolving images with a filter bank to generate filter responses.
- The MR8 filter bank consists of 38 filters but only 8 filter responses.
- The filter bank contains filters at multiple orientations but their outputs are ''collapsed'' by recording only the maximum filter response across all orientations. This technique achieves rotation invariance.
- The filter bank is shown opposite and consists of a Gaussian and a Laplacian of Gaussian (these filters have rotational symmetry), an edge filter at 3 scales and a bar filter at the same 3 scales.



Using MR8 Textons

- **The 24 element vectors are clustered to compute a Bag of Words of size 10**
- **Similar to the color textons described earlier, frequency histograms are computed for each image quadrilateral. These are the feature vectors used for classification**

Training Images



Apply MR8 filter bank on each pixel for L,a,b channels
Obtain 8 x 3 filter responses = 24 element vector

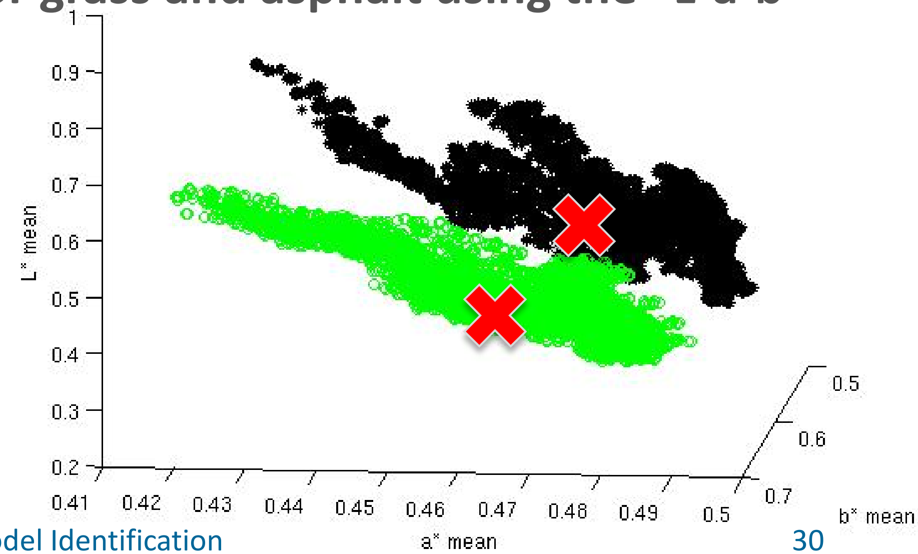
Table Of Contents

Feature Classification

- We use a generative algorithm known as Gaussian Mixture Models for learning. This algorithm imposes the assumption that each terrain class has been sampled from a Gaussian distribution.
- Using the K-Means algorithm we find the mean and variance of these Gaussians. Therefore for 3 different Terrain types we compute 3 clusters. For each feature vector, a probability is computed with which it would have been sampled from either of the terrain types. The class with the highest probability wins the class assignment.
- Below is an illustration of the data for grass and asphalt using the "L-a-b" feature vector.

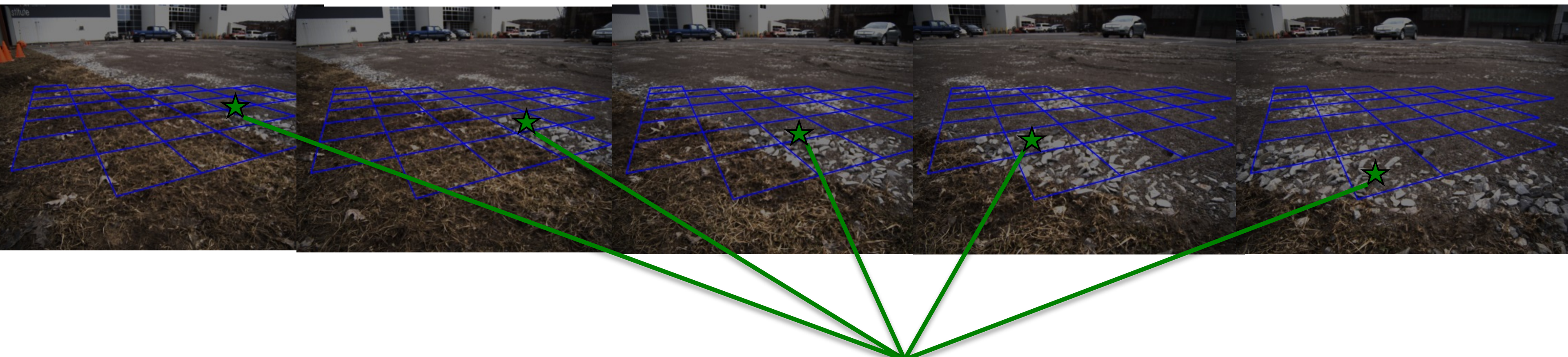
Feature Vector:

L* mean	a* mean	b* mean



Weighted Class Selection

- Once the distribution parameters (mean, variance) for each terrain type have been computed, each feature vector (for an image quadrilateral) is matched to the closest distribution.
- We calculate distance between the L-a-b feature vectors using Mahalanobis distance. The distance for the texton histograms (color, MR8) are calculated with the Chi-Squared metric.



Number of Views x	1. Matched Cluster Center 2. Distance to Center 3. Sample size (w)

Table Of Contents

Weighted Class Selection

- A particular grid cell on the ground can be viewed in multiple images. Each image quadrilateral (view) has its own belief of what class a particular terrain patch belongs to.
- We use a weighted voting mechanism to finalize best class match. Each image quadrilateral casts a vote between [1, m] of its belief, where m is the number of classes
- Votes are weighted based on two parameters:
  - (i) Size (in pixels) of image quadrilateral,
  - (ii) inverse of distance to nearest cluster center.

Accumulator

1	…	…	K

For all views of a grid:

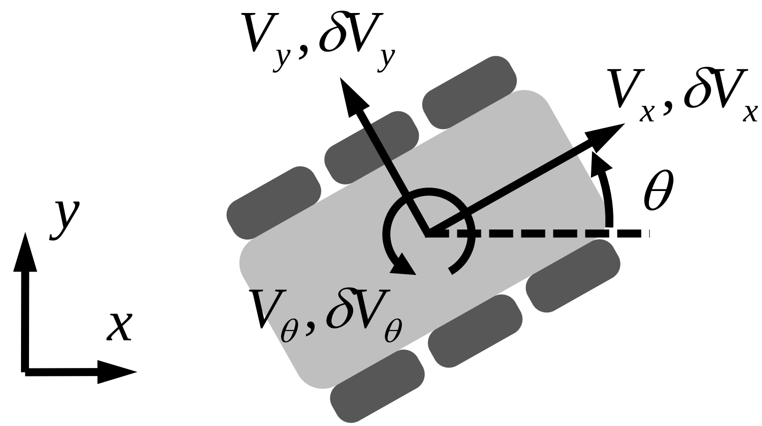
accumulator[matched_center] = projection_size/distance_to_cluster;

match = argmax(accumulator);

Table Of Contents

Slip Model

- **Velocity driven model**
- **Unconstrained kinematic differential equation**
- **Barring ballistic motion, this is the <u>general case</u>.**



pose rate (ground-fixed frame)

velocities (body frame)

velocity error (i.e. wheel slip)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} c\theta c\beta & c\theta s\beta s\gamma - s\theta c\gamma & 0 \\ s\theta c\beta & s\theta s\beta s\gamma + c\theta c\gamma & 0 \\ 0 & 0 & \frac{c\gamma}{c\beta} \end{bmatrix} \left(\begin{bmatrix} V_x \\ V_y \\ V_\theta \end{bmatrix} + \begin{bmatrix} \delta V_x \\ \delta V_y \\ \delta V_\theta \end{bmatrix} \right)$$

$$\gamma = roll, \qquad \beta = pitch, \qquad \theta = yaw$$

Slip Model

- **The model for calculating slip velocities in (x,y, yaw) is**

$$\delta V_x = a_1 V_x + a_2 |V_\theta| + a_3 V_x |V_\theta| + a_4 g_x$$
$$\delta V_y = a_5 V_x + a_6 V_\theta + a_7 V_x V_\theta + a_8 g_y$$
$$\delta V_\theta = a_9 V_x + a_{10} V_\theta + a_{11} V_x V_\theta + a_{12} g_x + a_{13} g_y$$

Symbol	Meaning
$a_1 \cdots a_{13}$	**Model Coefficients**
V_x, V_y, V_θ	**Linear and angular nominal velocity**
$\delta V_x, \delta V_y, \delta V_\theta$	**Linear and angular slip velocity**
g_x, g_y	**Components of gravity**

Learning Mobility Characteristics

- We learn the coefficients a_i for the visually classified terrain using a non-linear iterative least squares fit (Gauss-Newton algorithm).

- The mobility data is divided into a training set (700 seconds) and a test set (200 seconds).

- A mobility model is learned for each individual terrain type with data within the training set.

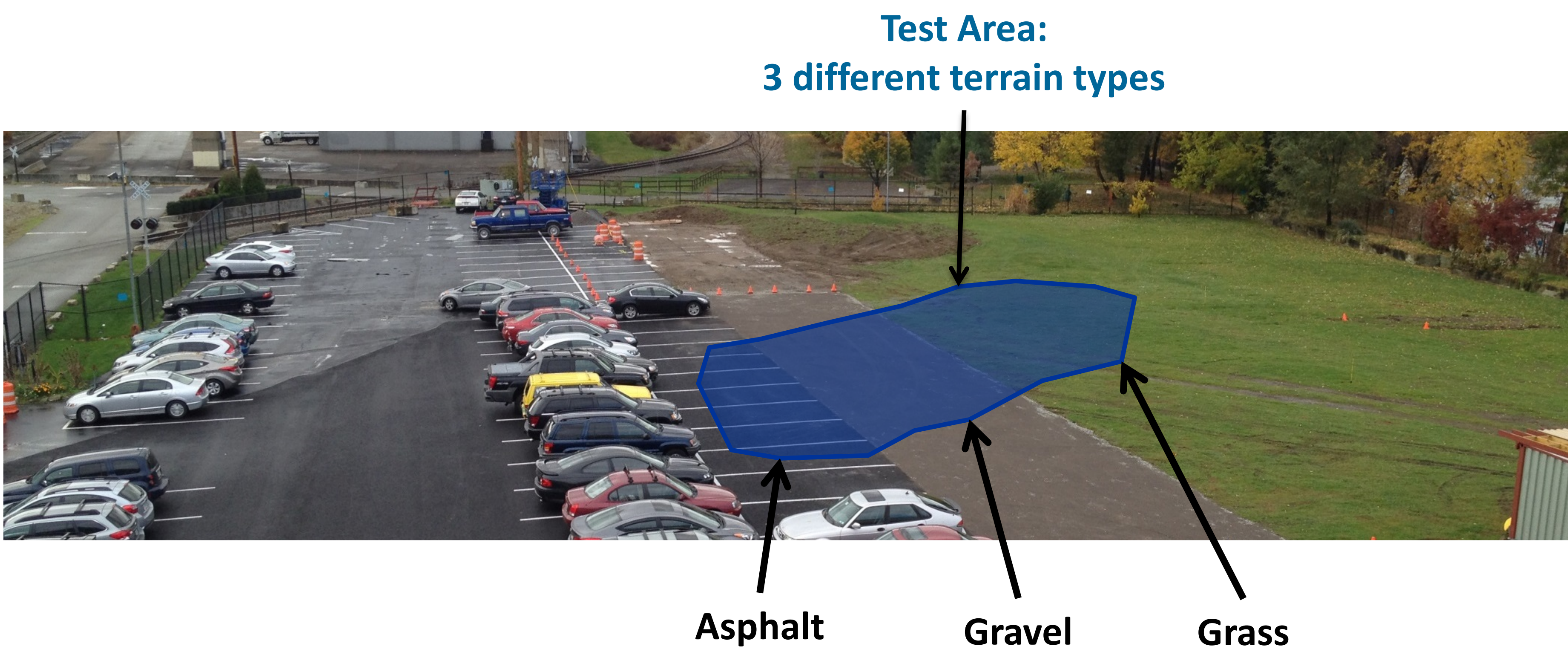
- For each terrain class in the test set, the respective mobility model is applied and the position estimation error is computed.

Table Of Contents

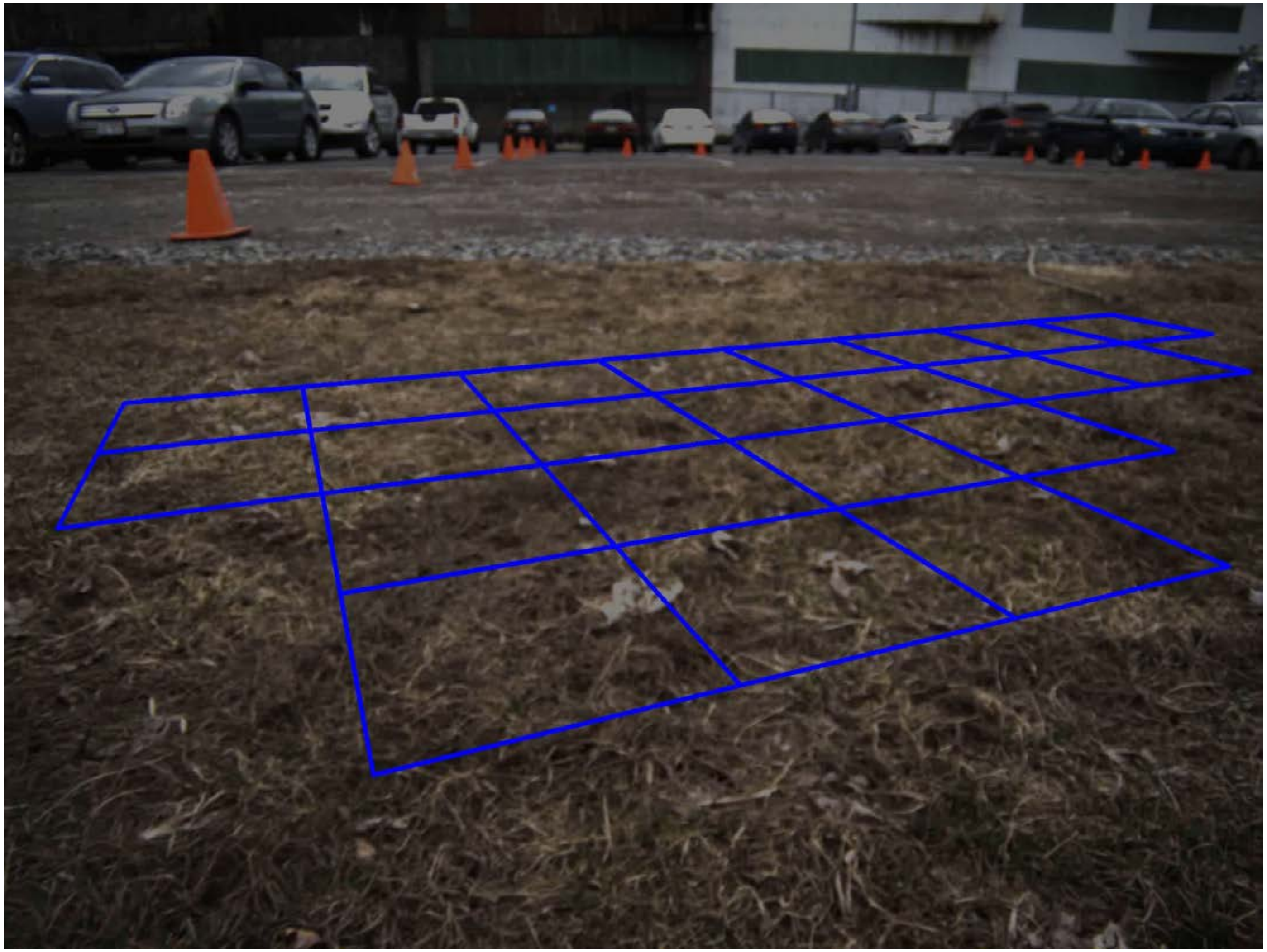
- Statement of Problem Studied
- Summary of Most Important Results
  - Perception aided slip prediction
    - Terrain Projection
    - Feature Generation
    - Classification
    - Class Selection
    - Learning Mobility Characteristics
    - <u>Experimentation Setup</u>
    - Experimental Results
  - Model Predictive Control
- Metrics for Reporting Period
- Conclusion / Outlook

Experimentation Test Site

- The test area presents four different types of terrain transitions.



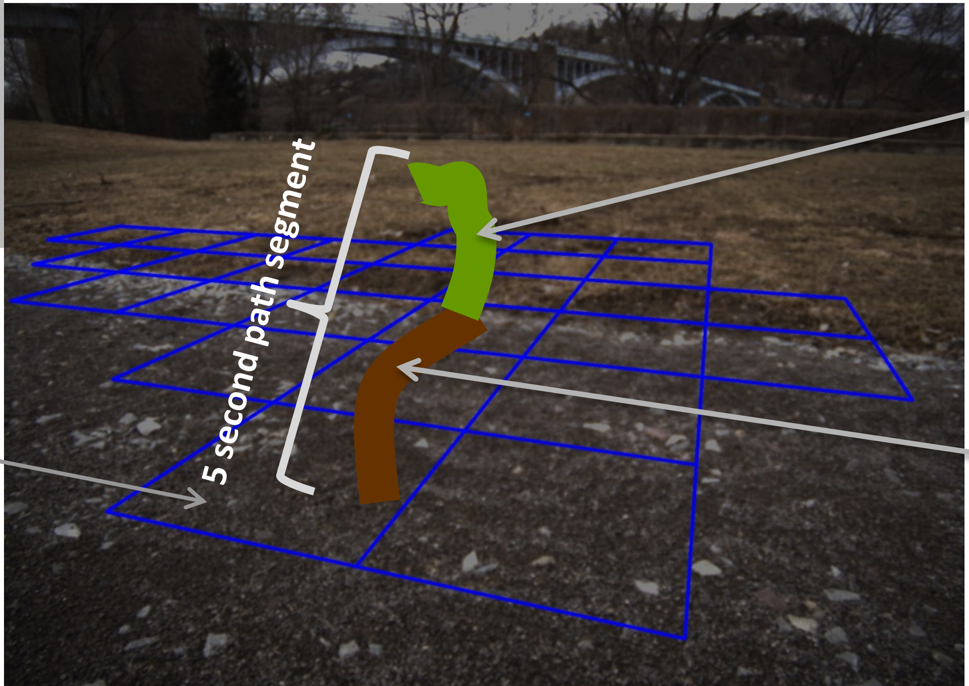
Data Collection (Video)



Experimental Setup

- For this experiment we used 3 different terrain types.
- An average model is also learned. It treats the entire area as if it was homogeneous.
- To gauge perceptual cueing performance, we apply each slip model to a 5 second vehicle trajectory. As shown in the figure, a second slip model is applied right after the transition if perception said a transition was going to occur.
- The entire vehicle path (several meters) is split into 5 second trajectories and slip is repeatedly predicted for each 5 second trajectory.

The basic idea of perceptual cuing across a Terrain Transition



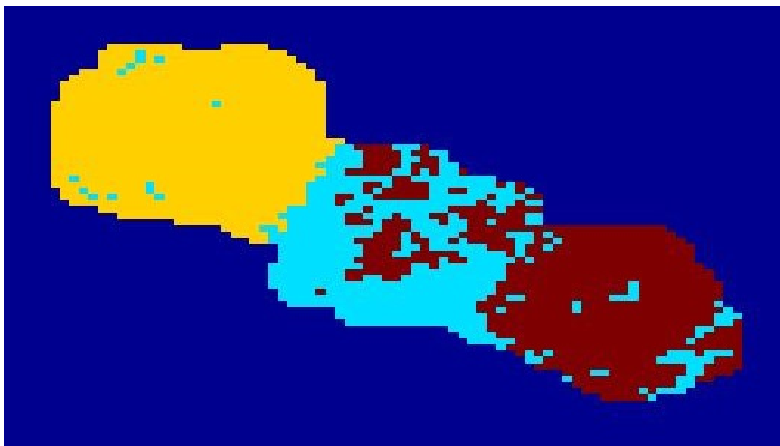
Apply Grass Mobility Coefficients

Apply Flat Ground Mobility Coefficients

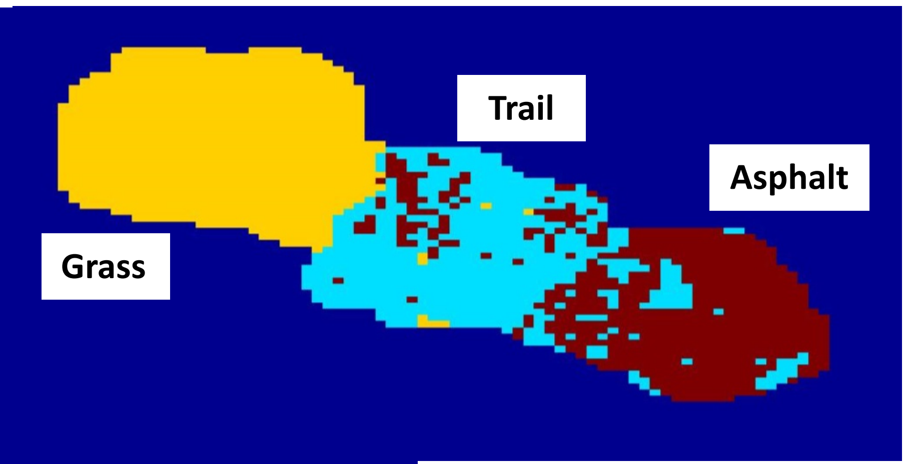
Each quadrilateral in image is a 30cmx30cm grid on the ground

Perception Results:
Classified Terrain Maps

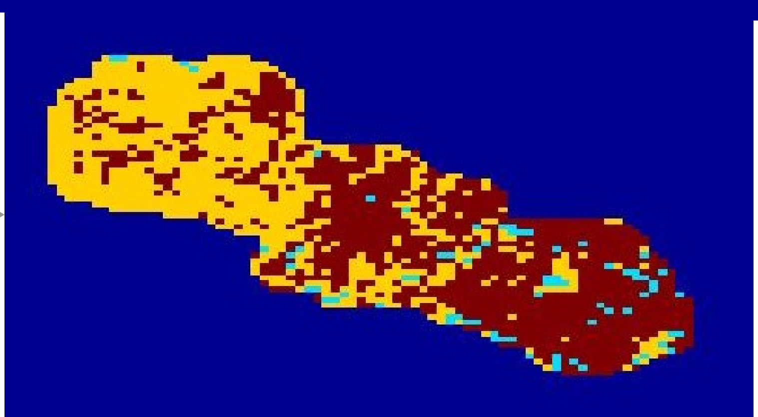
- Comparing the terrain classification results with the imagery of the experimentation site, we see that the textons perform much better than simple color statistics. This is expected given the importance of texture of the terrain over simple color.
- However we can only analyze this qualitatively since we perform unsupervised classification. This is a more practical approach for easily deploying on the field, because it does not require creating a hand labeled dataset to train on.



MR-8 Textons



Color Textons



Color Statistics

Pose Instrumentation

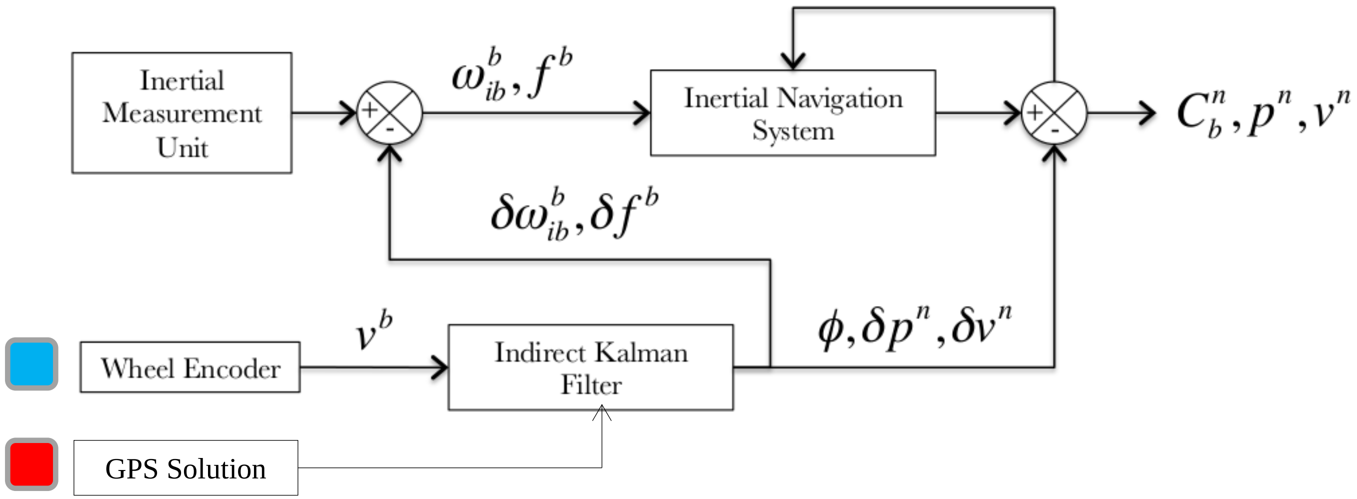
- Terminology: Pose = Position + Orientation.
- The system relies on a pose estimate both during training and during operation.
- We need a pose solution to be able to span the time from when terrain is seen to when it is touched by the wheels.
- The vehicle is not a point; the camera is not where the pose box is; neither is where a particular camera pixel lands on the ground. Hence attitude needs to be known too. Attitude data allows us to determine precisely which camera pixel hits the ground where.

Pose Architecture

We compute two solutions in our pose estimation system:

Local Pose: Odometry + Inertial

Global Pose: GPS + odometry + Inertial



- $\mathbf{x}^a$ Vector $\mathbf{x}$ with components written in the a coordinate frame.
- $\mathbf{f}$ Non-gravitational acceleration (Also known as specific force) as measured by an accelerometer.
- ω Angular rate vector.
- ω_{ie} Angular rate vector of frame e relative to frame i.
- $\mathbf{C}_b^n$ Rotation matrix (Also known as direction cosine matrix) relating frame b to frame n. When pre-multiplied with a vector having components written in frame b it produces the same vector with components written in frame n.
- $[\mathbf{x}\times]$ Skew symmetric matrix form of vector $\mathbf{x}$ such that $\mathbf{x}\times\mathbf{y} = [\mathbf{x}\times]\mathbf{y}$.
- $\hat{\mathbf{x}}$ Measurement or model of $\mathbf{x}$ including true signal and additive error.

Experimental Design : Dual Pose

- We will show that neither the training nor the operational system relies on highly accurate pose, or on a truly global (georeferenced) position estimate. We know this because we tested with two ground truth pose solutions.
- Testing with dual poses allows us to assess deployment issues with less capable instrumentation.
- A system reliant on RTK GPS would not be very practical and we show it to be unnecessary.

Experimental Design : Contradiction

- It is apparently contradictory to use a pose system corrupted by wheel slip (in the odometry aiding) in order to calibrate an odometry slip model. How does a broken sensor calibrate itself?
  - However, a good gyro (present in both local and global pose solutions) renders angular slip perfectly observable and our results have always shown this to be the dominant slip behavior.
  - So while our pose solution is corrupted by wheel slip, the fused solution is <u>not corrupted much</u> because it is the gyro that is really doing the measuring that matters. The function of odometry in the fused solution is to damp accelerometer drift.
- That leads to the question of why we need to predict slip at all if we can measure it. Response:
  - We can only measure angular slip but translation slip is correlated to it so a slip model <u>makes both observable</u>. That correlation model is of extreme value in inertial navigation because translation is the dominant source of remaining error in pose estimation.
  - You cannot measure the future. In true prediction problems, gyros are useless and you are forced to use a model. We are able to produce a predictive model from experience gained during those times when you could measure it.

Pose Solution Quality

- Pose error is furthermore somewhat exacerbated in our experiments by extreme driving patterns which introduce more wheel slip. We often used extreme driving patterns to maximize signal-to-noise and to stress test our conclusions. However, the pose system can handle this for reasons given in last slide. Specific performance numbers below.
  - Heading Error: In both local and global pose, our tactical grade gyro exhibits an angular error drift of a mere 10 deg/hour[1] because both solutions use a gyro for heading.
  - Global Pose Translation Error: The GPS aided pose solution has 1 cm or so absolute position error.
  - Local Pose Translation Error: The local pose solution has a translation error of 1% over every 100 meters if driven in a straight line.
- IOW, for driving 1 meter to reach the ground in view, the 1 cm of error we experience (in correlating a camera pixel with the spot under the wheel), is negligible compared to the benefit of cueing a model transition that affects up to 5 meters of subsequent prediction.

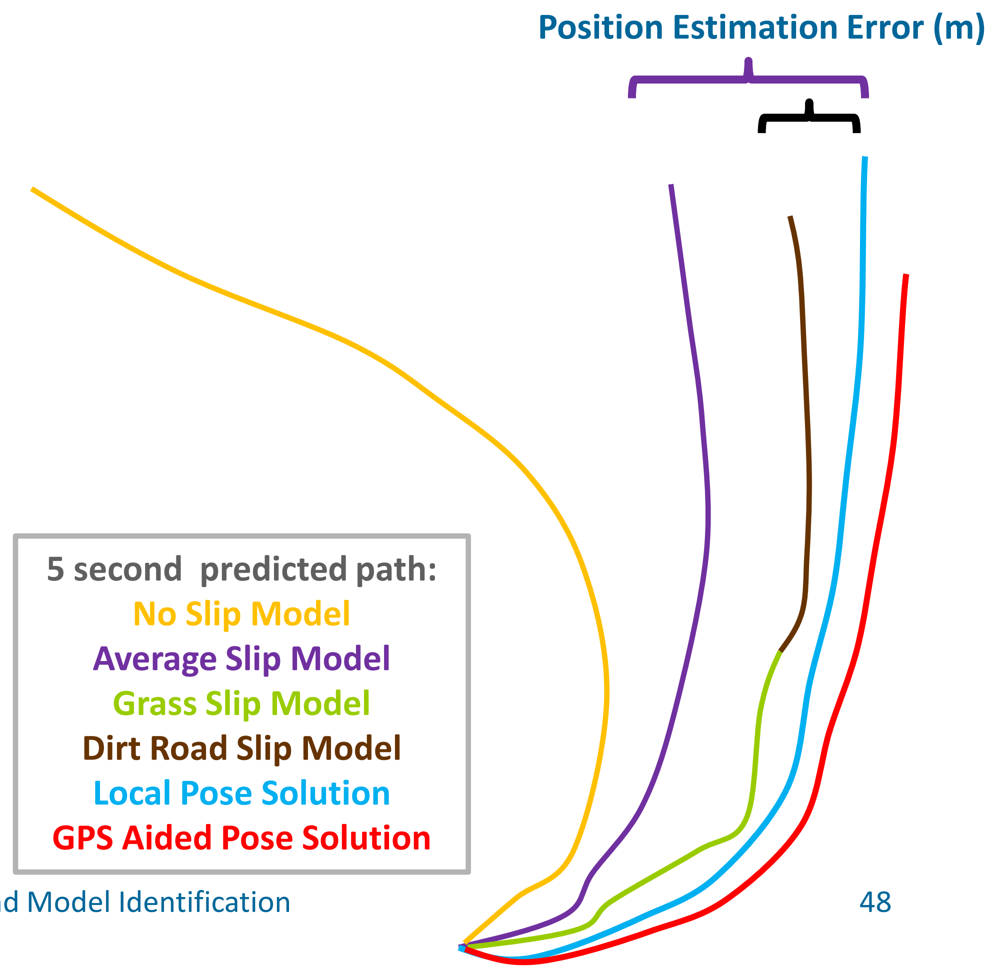
1: "Mere" because we care about the angular error that accumulates over the second or two it takes to drive over the terrain in view. 10 deg/hr is 50 MICRO rads per second.

Distinguishing Slip Signals

- This graphic illustrates the slip analysis we perform using perceptual cueing. Each curve is a path simulated by integrating velocity measurements over 5 seconds. As described earlier, the integration velocity is determined as

  V_int = V_nom + V_slip.

- We iteratively run a sliding window through the data to get the next path. The shift of the window is 1 second. Hence for a 5 second path, a given segment will appear at most 5 times
- The pose and velocity solution data is at 100Hz. Therefore, each path segment consists of 500 data points.



Using the Slip Model

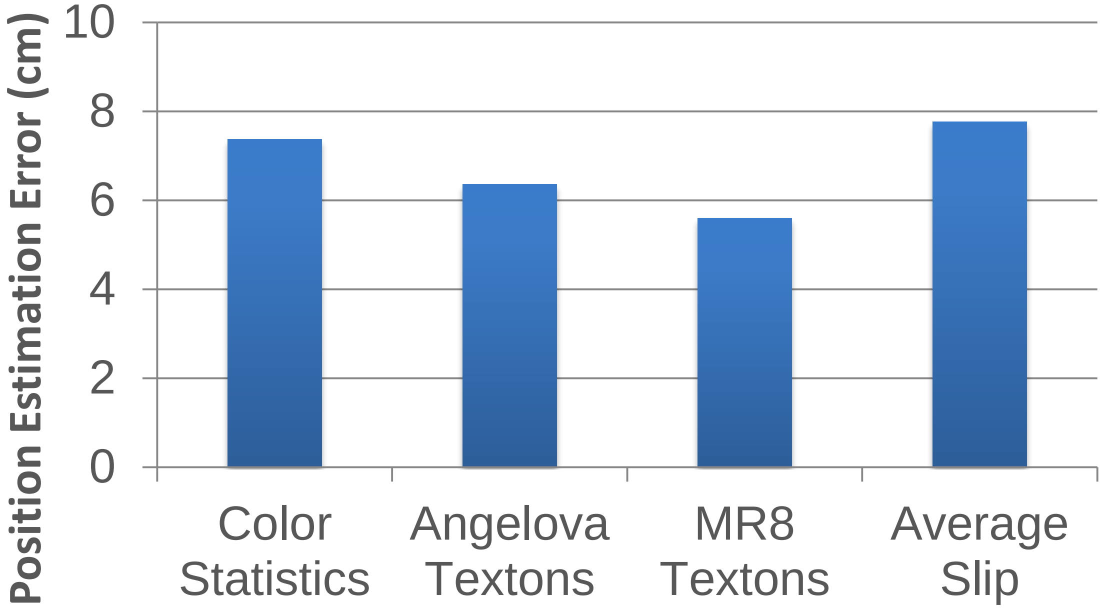
- The nominal velocity (V_nom) is the velocity computed from wheel rotations and a no-slip assumption represented in vehicle frame. This term is corrected for slip by adding a V_slip term computed by a calibrated kinematic model.

- The curve without a slip model is computed with V_slip = 0. As is evident in the cartoon, the difference in angular velocity results in a large position prediction error over time.

- The perceptually cued model swaps the slip parameters applied in forward integration based on a perception prior.

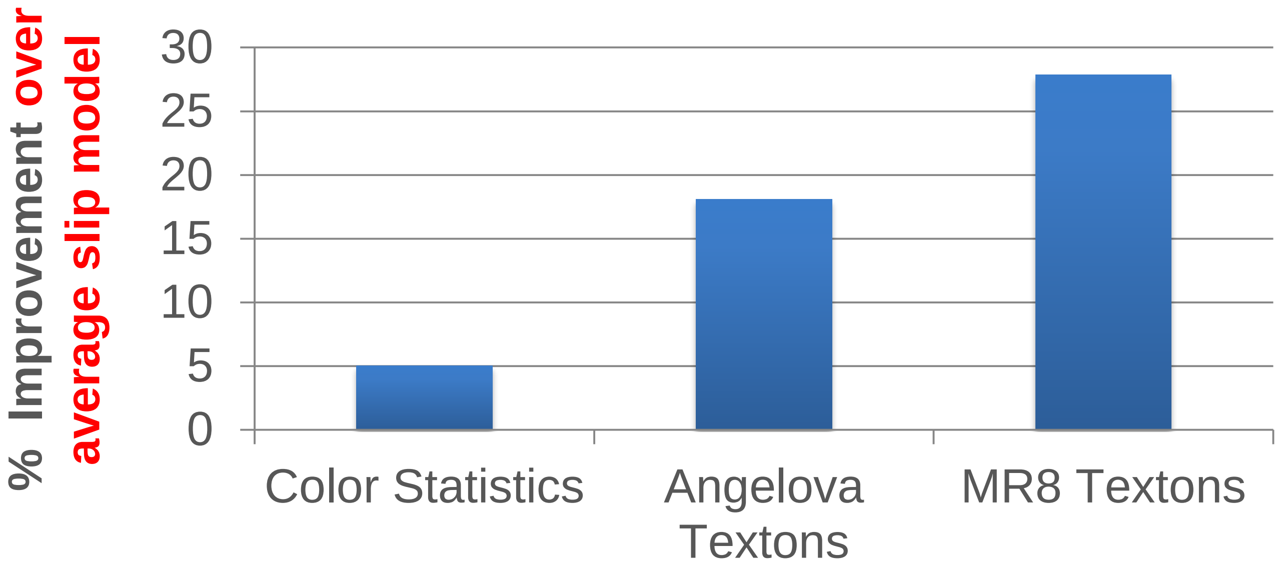
Table Of Contents

Position Estimation Error: Local Pose



- **At a top speed of 1m/s, the Husky would've travelled 5 meters. 8 cm is significant estimation error even with a well calibrated average model. We show that this can be greatly reduced by incorporating a perception prior (perceptual cueing).**
- **The estimation error above is the average error (illustrated in the graphic) of the end points of paths computed from slip-compensated odometry vs the pose system "ground truth" path.**
- **Since this is computed using a local pose solution, the estimation error is caused primarily due to angular slip (which is usually the most dominant).**

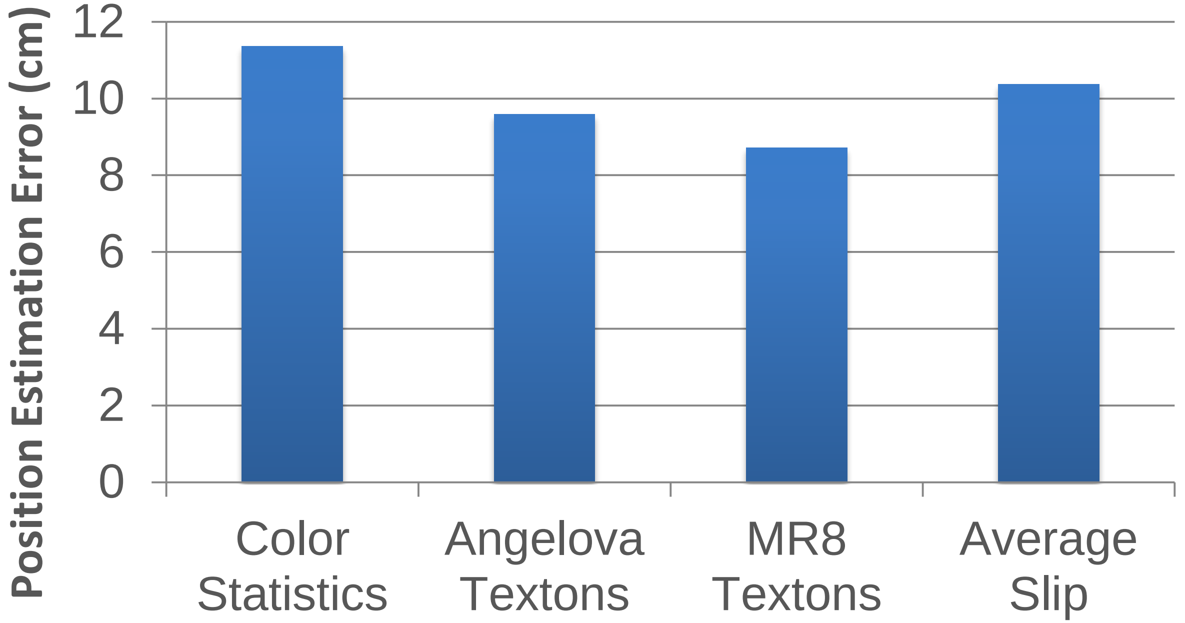
Position Estimation Error: Local Pose



- Same data as graph on last slide.
- This graph is simply the percentage improvements (reduction of estimation error) for the perceptually cued slip models over an average model.
- For example using the numbers in the previous graph:
    - % improvement = 100 * (Average_Slip_Error – Color_Statistics_Error) / Average_Slip_Error

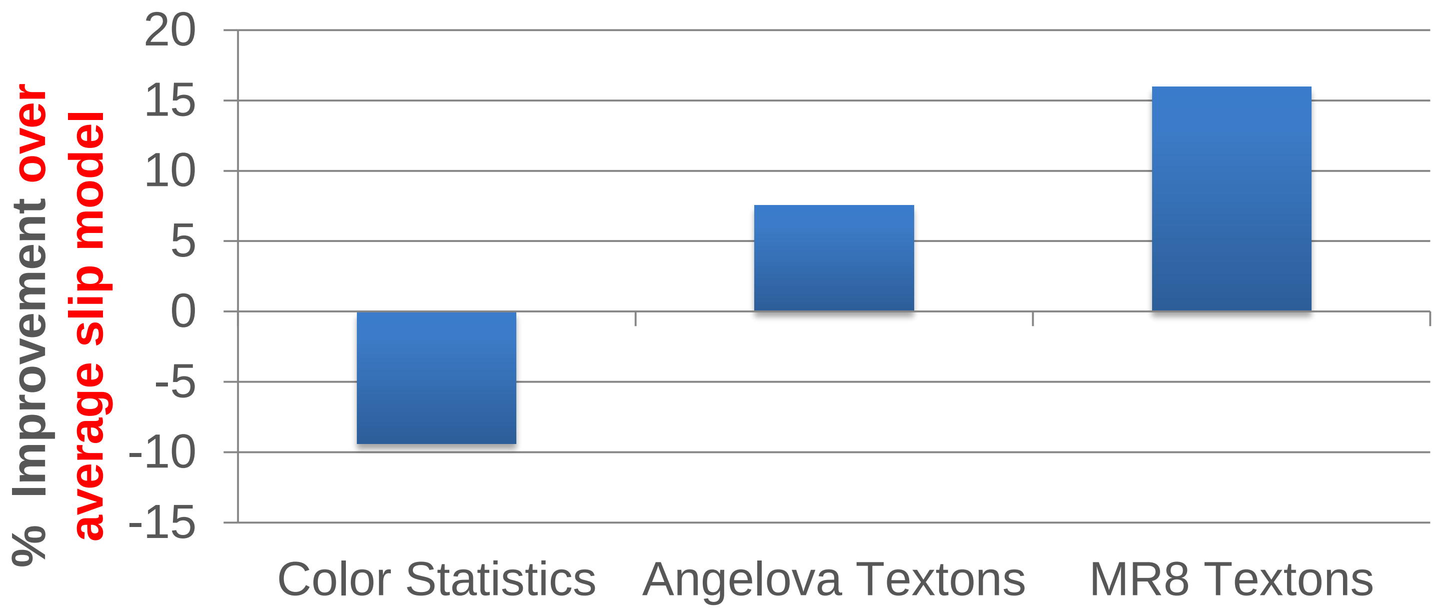
Position Estimation Error:

Global Pose



- **The above graph is generated using a global pose solution, that is aided with GPS data of accuracy within 2cm. Therefore this solution can be considered as ground truth.**
- **As described earlier, the global pose solution has minimal translational error so it allows us to observe such error in the slip-compensated odometry solution.**
- **Hence the estimation error for slip-aided odometry using global pose is higher than that for local pose.**

Position Estimation Error: Global Pose



- The percent improvement over an average model is computed as described previously.
- We see that the color statistics model performs worse than an average model. <u>This is because the terrain classification with this feature set is actually not very good as we have already seen</u> so it simply means we should not use this feature set.
- As suggested on slide 45, the estimation-error-of-both-average-and-cued-models is larger when global pose is used to measure slip. However the <u>difference between</u> estimation-error-of-average-and-cued-models is essentially the same, so the percent improvement is reduced. For MR8 Textons, the local pose calculation is (8-5)/8 = about 30%. For global pose it is (10-8)/10 = about 20%. So this is the same conclusion but from a better measurement.

Perceptual Cuing Results

- We have shown a measureable improvement in prediction error for a 5 second prediction horizon with respect to an average model that does not distinguish terrain types.
- The improvement magnitude is ~28% for MR8 textons and ~18% for color based features for a local pose ground truth solution. The improvement is slightly lower for a global pose ground truth solution.
- In both cases, more clusters (terrain classes) performs better. This might however lead to overfitting and perhaps not generalize well.
- More extreme terrain transitions (like asphalt to gravel) are likely to see higher prediction improvements.
- <u>We have demonstrated with high confidence that slip prediction can be improved with perception</u>.

Table Of Contents

Model Predictive Control

- In year 2012, we showed the effects of slip on open-loop Model Predictive Control

- This year, the goal is to show the effects of slip on Model Predictive Control approach to path following. This is a more realistic scenario as every fielded ground robot runs on closed loop control.

- Initially this was developed in simulation and we have transition this to a real vehicle in this reporting period.

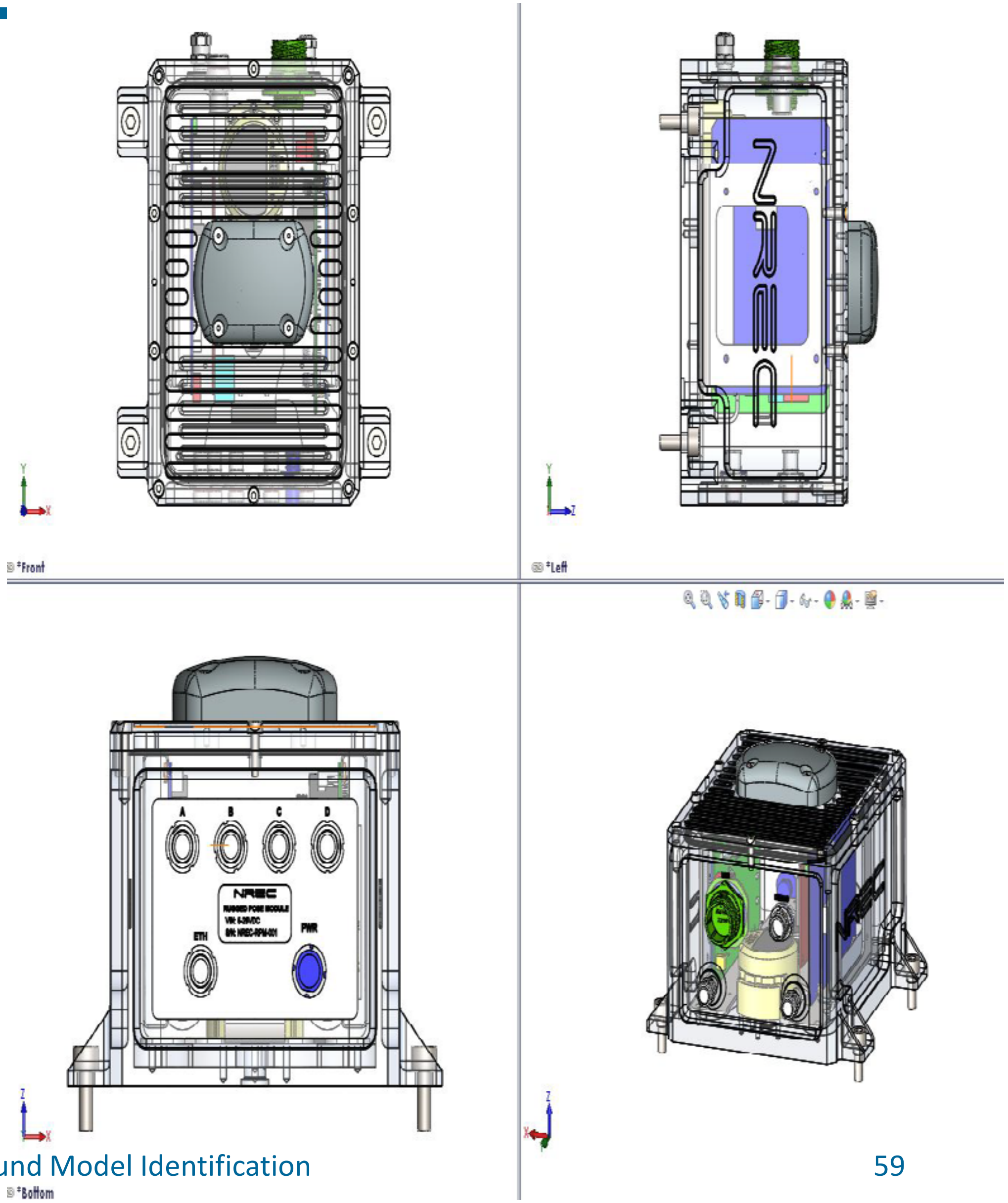
- The vehicle is required to “follow” a desired path adhering to the following constraints
  - The generated trajectory should take into account the following constraints: path smoothness, velocity (linear and angular) limits, lateral and longitudinal acceleration limits.

MPC - Platform

- Our platform was Husky, a 4x4 skid-steered, all-terrain, all-electric vehicle built by Clear Path Robotics LLC.
- We retrofitted the platform with wheel encoders, a pose system and a (forward looking) AVT Prosilica GigE camera

MPC – Pose system

- IMU for the pose system is a Honeywell HG1930 with MEMS based accelerometers and gyros (drift rate of about 10 degrees per hour)
- A NovAtel OEM V-3 GPS receiver card along with an Antcom G5Ant GPS antenna is used for receiving GPS measurements
- Two wheel encoders on the aft wheels provide raw odometry measurements
- A "smooth", "local" pose solution is generated using the inertial measurements and wheel encoder measurements at 100Hz for the controls
- "Ground truth" data is generated by post-processing the "rover" GPS and inertial data with the "base" station's GPS observations



MPC – Problem Formulation

- We pose the task of model predictive path following as an optimization problem
- The following equation describes the form of the objective function we are looking to optimize

$$J = \frac{1}{2} \sum_{\substack{k=1 \\ ArgMin(u)}}^{n} \left((x_k^T\, Q x_k) + (u_k^T R u_k) \right)$$

  - J is the objective functional
  - n is the number of time steps to perform the optimization
  - x_k is the state at time step k of dimension (1xl)
  - u_k is the control effort at time step k of dimension (1xd)
  - Q is the l x l positive semi-definite weight matrix
  - R is the d x l positive definite weight matrix

- The state evolution (system dynamics) is given by the following equation

$$x_{t+1} = A_t\, x_t + B_t u_t$$

  - A_t is the linearization of the state x_t at time step t
  - B_t is the linearization of the control u_t at time step t

- We solve this optimization problem using a receding horizon, iterative LQR approach

MPC - State Vector

- The following fields make up the state vector at any given time
  1. δx – difference in longitudinal position (meters)
  2. δy – difference in the lateral position (meters)
  3. $\delta \theta$ – difference in the yaw (radians)
  4. $\delta \dot{x}$ – the rate of chance of difference in longitudinal position (meters per second)
  5. $\delta \dot{y}$ – the rate of change of difference in lateral position (meters per second)
  6. $\delta \dot{\theta}$ – the rate of change of difference in yaw (radians per second)
  7. v_d – the desired longitudinal speed (meters per second)
  8. k_d – the desired curvature (1/meters)

- The reference signal we are trying to follow is <u>always</u> the one generated by the pure pursuit follower. However, it is trivial to swap this out with a compatible trajectory generated by a different means (direct driving commands, model predictive trajectory generator, high level path planner etc.)
- In other words, we are not trying to show a better control algorithm. We are trying to show that whatever the reference trajectory is, you can follow it better by compensating for slip predictively in MPC. We are choosing whatever a slip free vehicle would do using a simple controller for our reference.

MPC - Approach

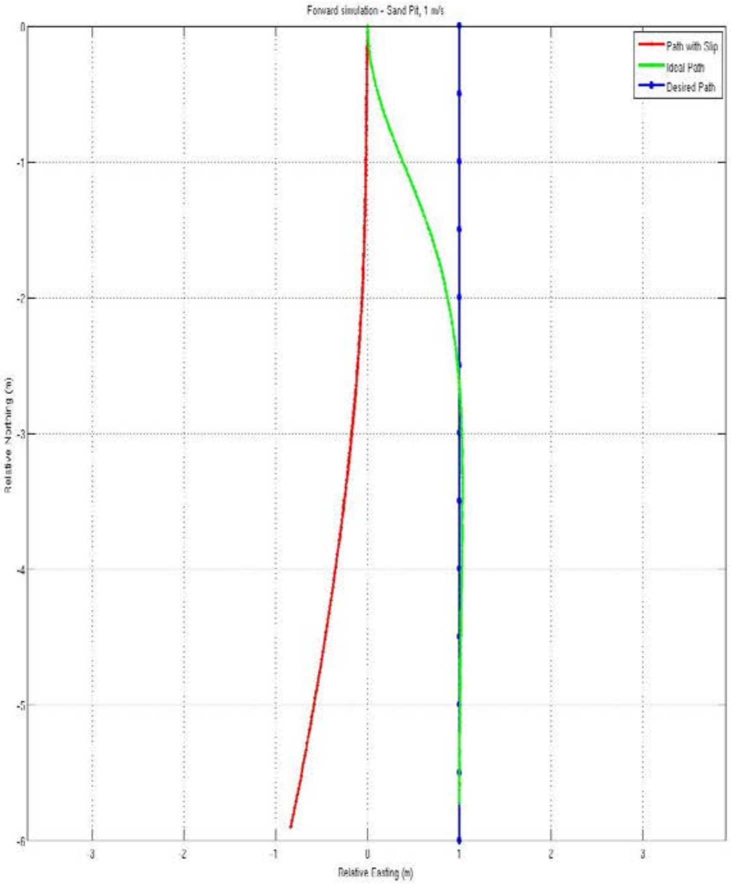
– We are given a desired path in the world to follow along with suggested speeds as well as the calibrated slip model of the vehicle and the terrain.

– Every iteration of the algorithm, we do the following steps

1. Localize the vehicle onto the path (find the point on the path that is closest to the vehicle)
2. Compute the look-ahead point on the path – this is a point on the path we are trying to steer the robot to
3. Forward simulate to predict the "no slip" path the robot would have traversed along with the set of controls <u>under "ideal" conditions – no slip</u>
4. Forward simulate to predict the "slippy path" path the robot would have traversed when executing the controls generated in step 3 <u>taking slip into consideration</u>
5. Compute the correction to be applied to the controls generated in step 3 using iLQR
6. Pass the adjusted controls to the vehicle controller

MPC – Inner working snapshot

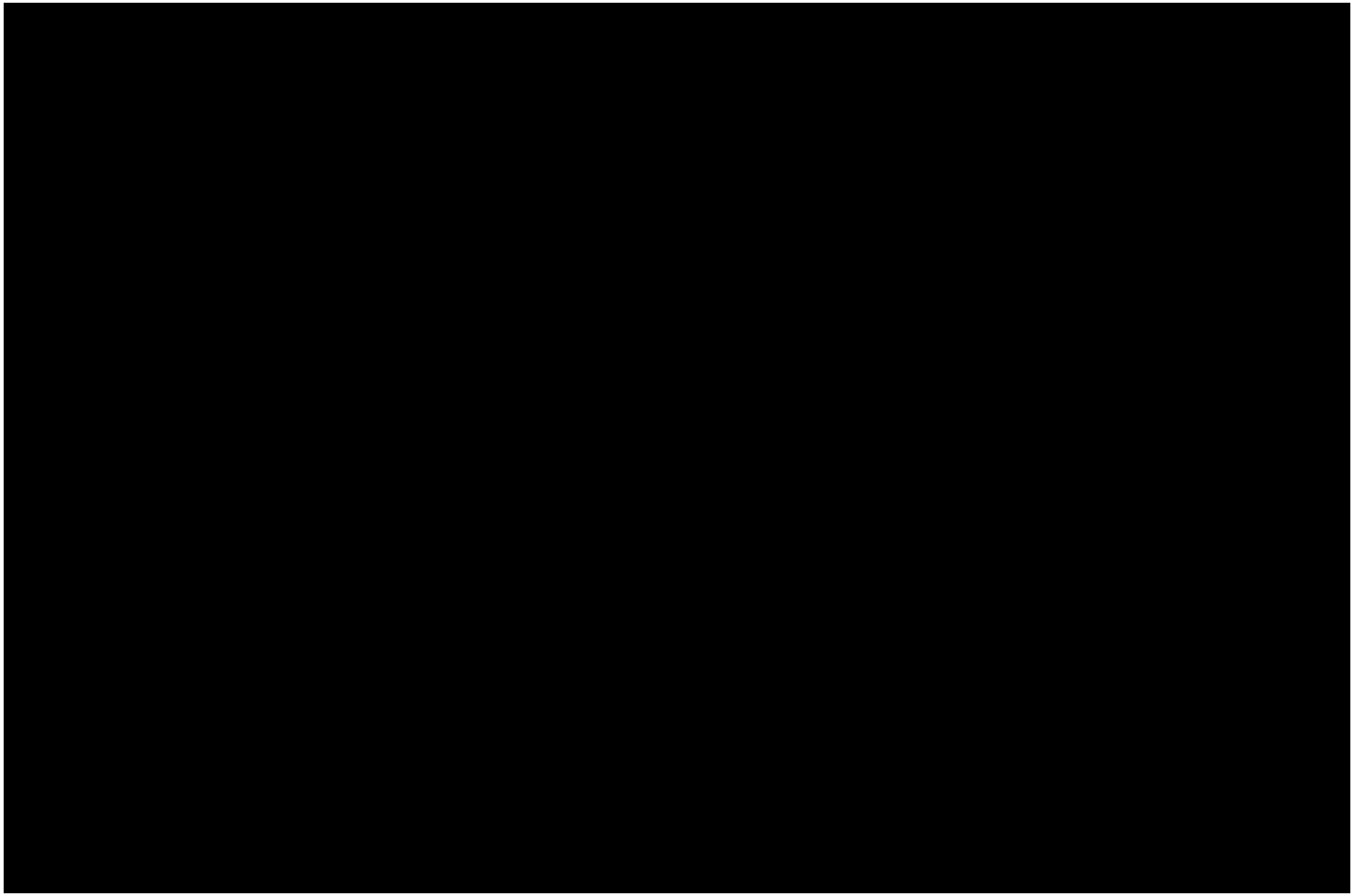
- **Blue line denotes the high level desired path starting at the top and going down**
- **Green line denotes the (reference) path the vehicle is predicted to traverse when slip is NOT considered**
- **Red line denotes the path the vehicle is predicted to traverse when slip is considered but MPC is off.**
- **MPC tries to minimize the errors between the green and the red paths**



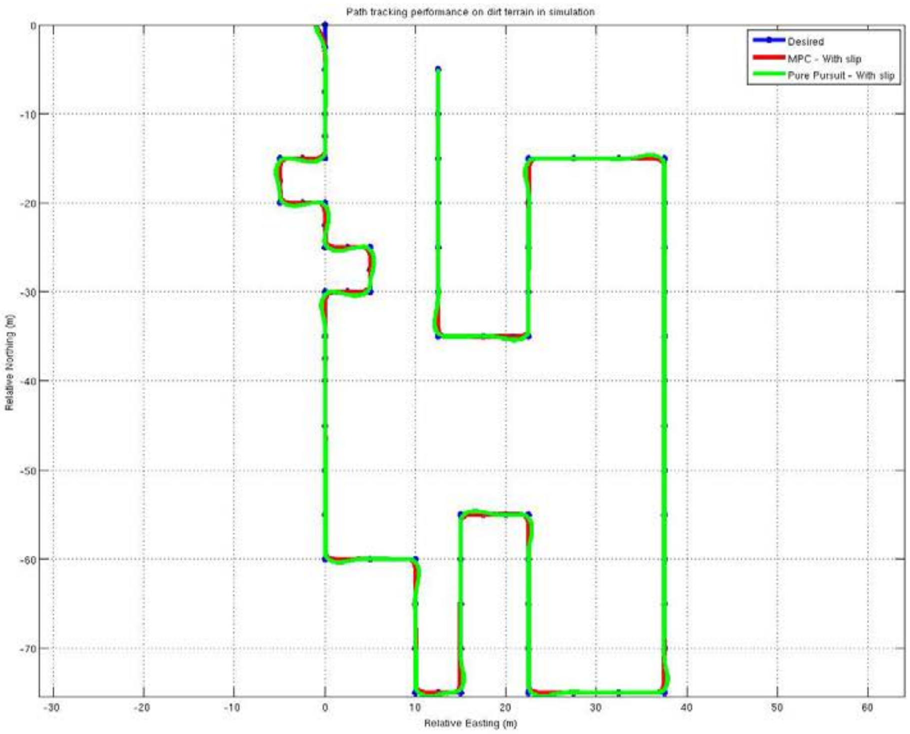
MPC – Trials in Simulation - Setup

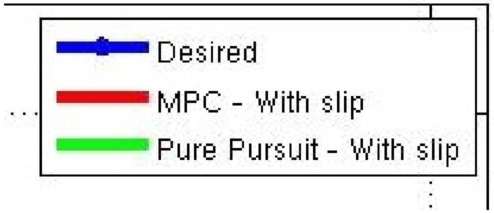
- Total path length 312.5 meters
- Twenty 90 degree turns placed at various distances apart with 8 right turns and 12 left turns
- Three scenarios of varying slip characteristics
  - Dirt (parameters learned from real world trial)
  - Sand Pit (parameters learned from real world trial)
  - Hard to turn left (parameters modified to simulate a desired behavior)
- Maximum mission speed of 1.0 m/s

Path Tracking in Simulation Video – 4X Speedup



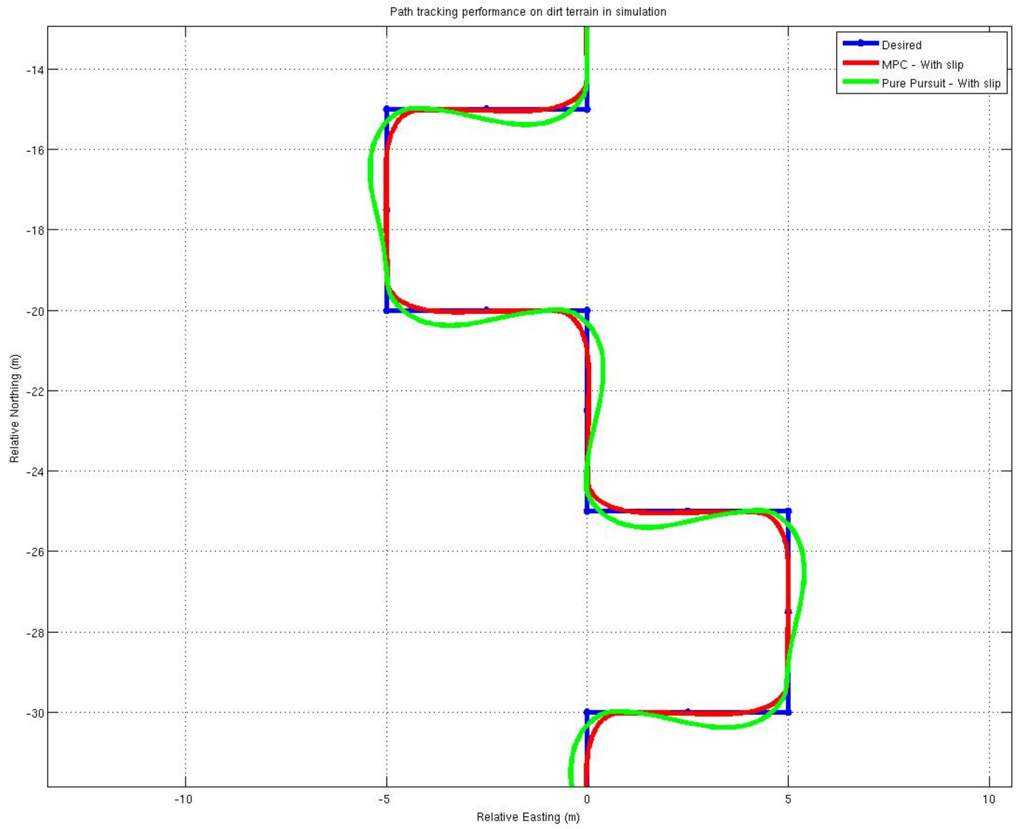
Simulated Performance in Dirt Terrain

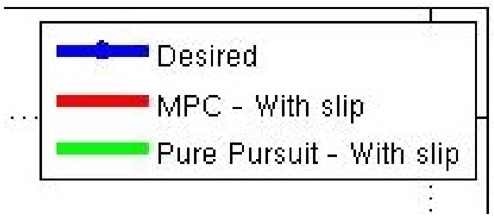




Algorithm/Performance	Pure Pursuit	MPC
Cross Track Error (max/mean) (m)	0.41/0.08	0.29/0.02

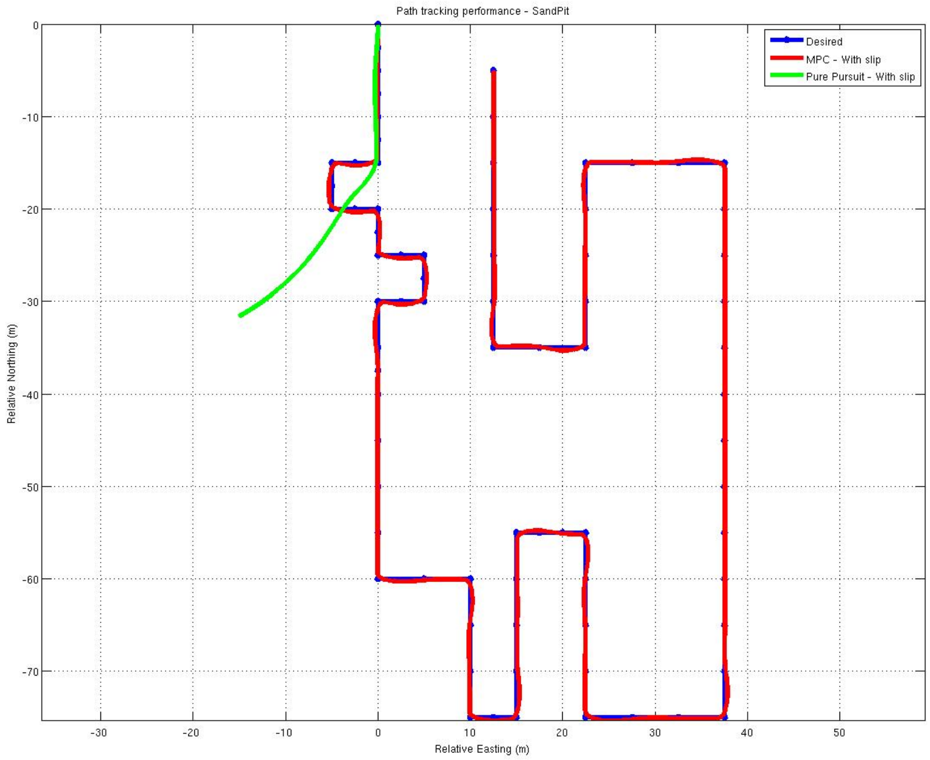
Simulated Performance in Dirt Terrain – A closer look

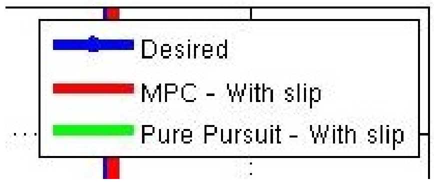




MPC performance is essentially perfect – within the pose estimation resolution

Simulated Performance in Sand Pit

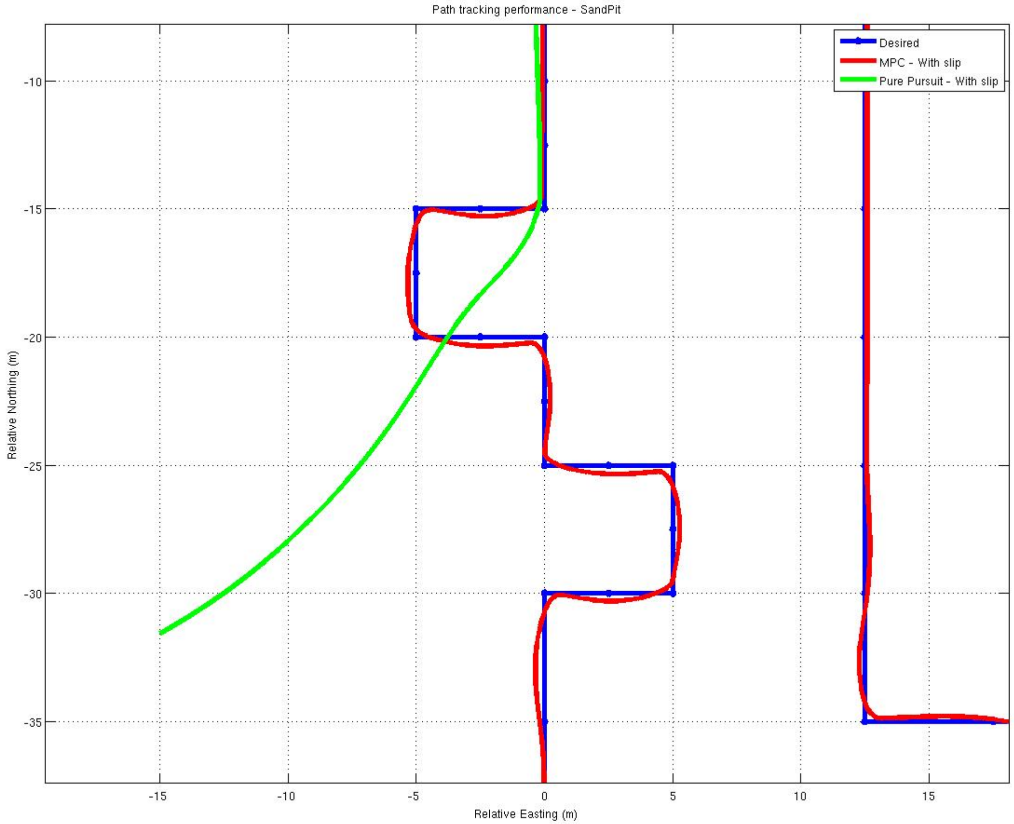


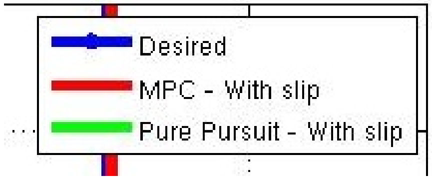


MPC performance is excellent. Pure pursuit fails utterly.

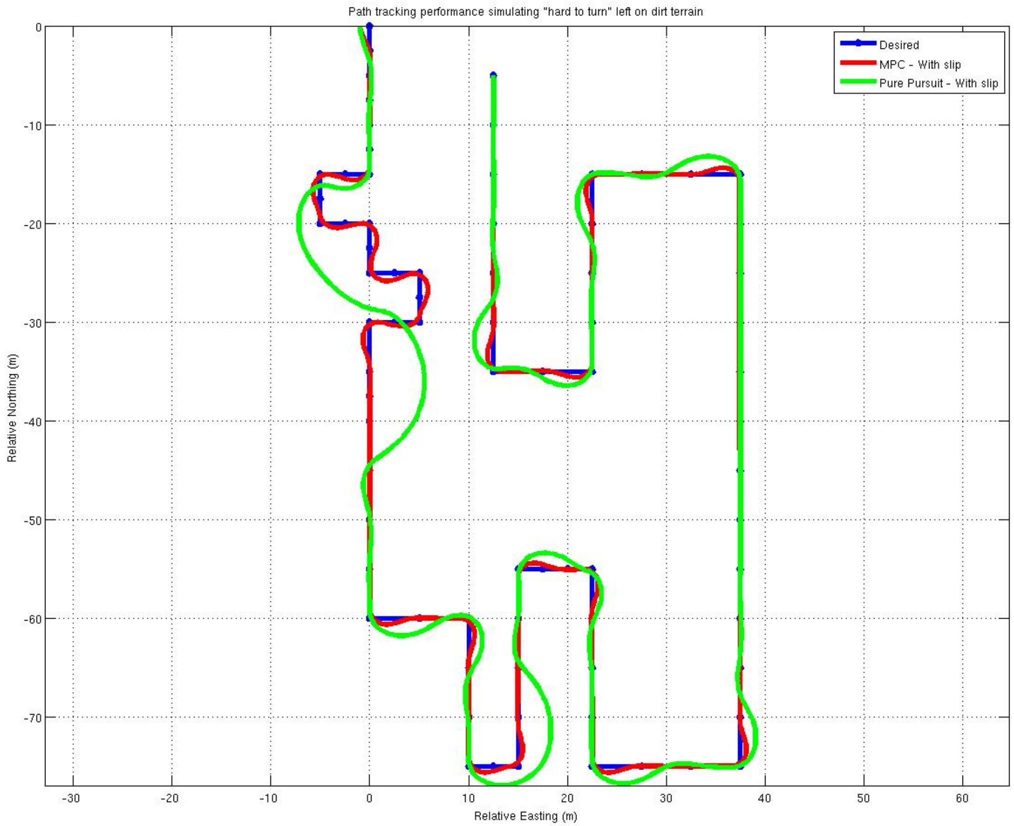
Algorithm/Performance	Pure Pursuit	MPC
Cross Track Error (max/mean) (m)	**13.45/10.60**	**0.38/0.14**

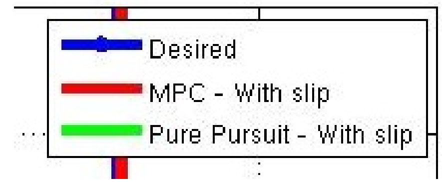
Simulated performance in Sand Pit – Zoomed In





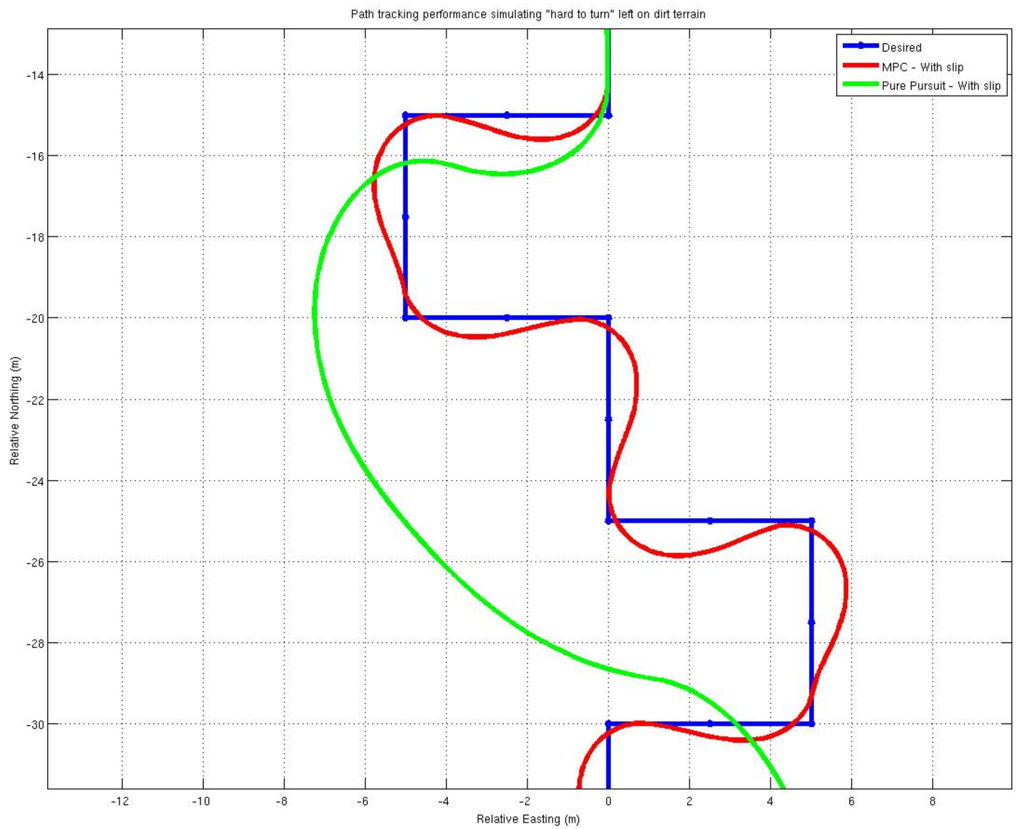
Simulated performance in a “hard to turn” left scenario

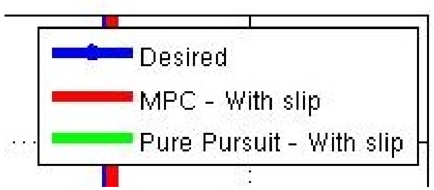




Algorithm/Performance	Pure Pursuit	MPC
Cross Track Error (max/mean) (m)	5.03/0.54	0.73/0.14

Simulated Performance in "hard to turn left scenario" – a closer look





This is analogous to a military robot adapting automatically to battle damage.

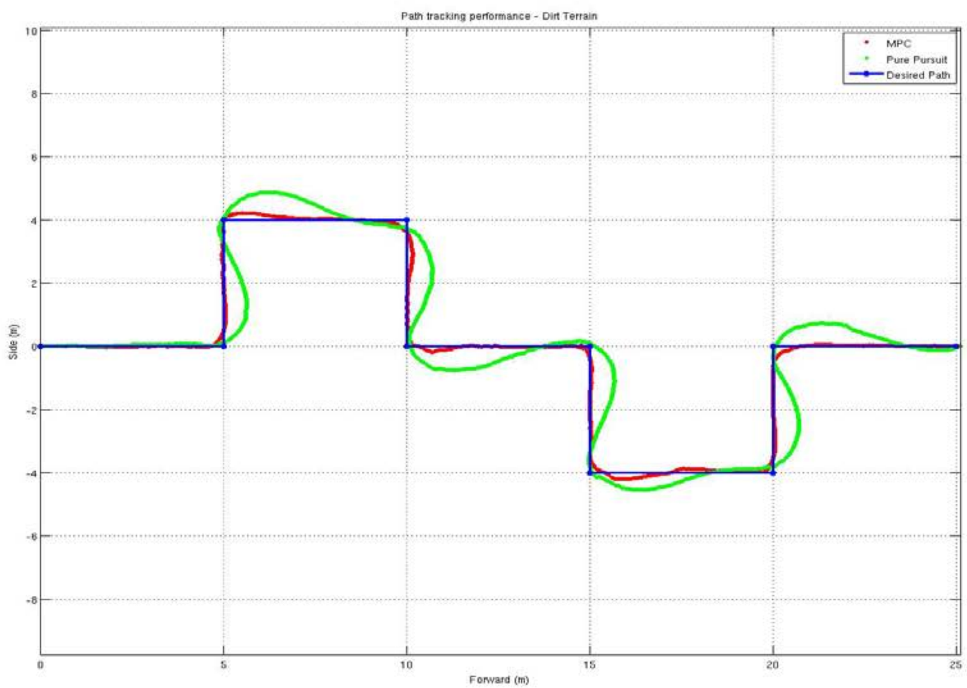
MPC Real World Trials Setup

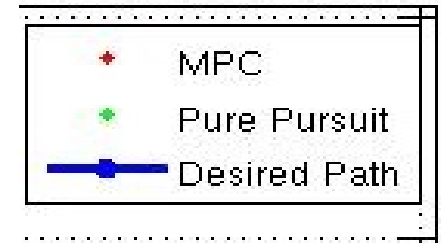
- Two terrains (dirt and sand pit) of varying slip characteristics were used for the trials
- Slip models were “learned” by driving the vehicle in the representative terrain similar to the perceptual cueing trials
- Due to space restrictions, path lengths were limited to about 25 meters but their shape were representative enough to show real capability

MPC Trials – Dirt Terrain

- The terrain was about 15 meters wide and 25 meters long
- Terrain consisted primarily of packed dirt with occasional loose rocks and tufts of grass
- Trials were conducted on a path with eight 90 degree turns (4 left and 4 right) spaced anywhere between 4 to 5 meters apart
- Total path length is 41 meters and max desired speed was 1.0 m/s

Path Tracker Performance on "Dirt Terrain"



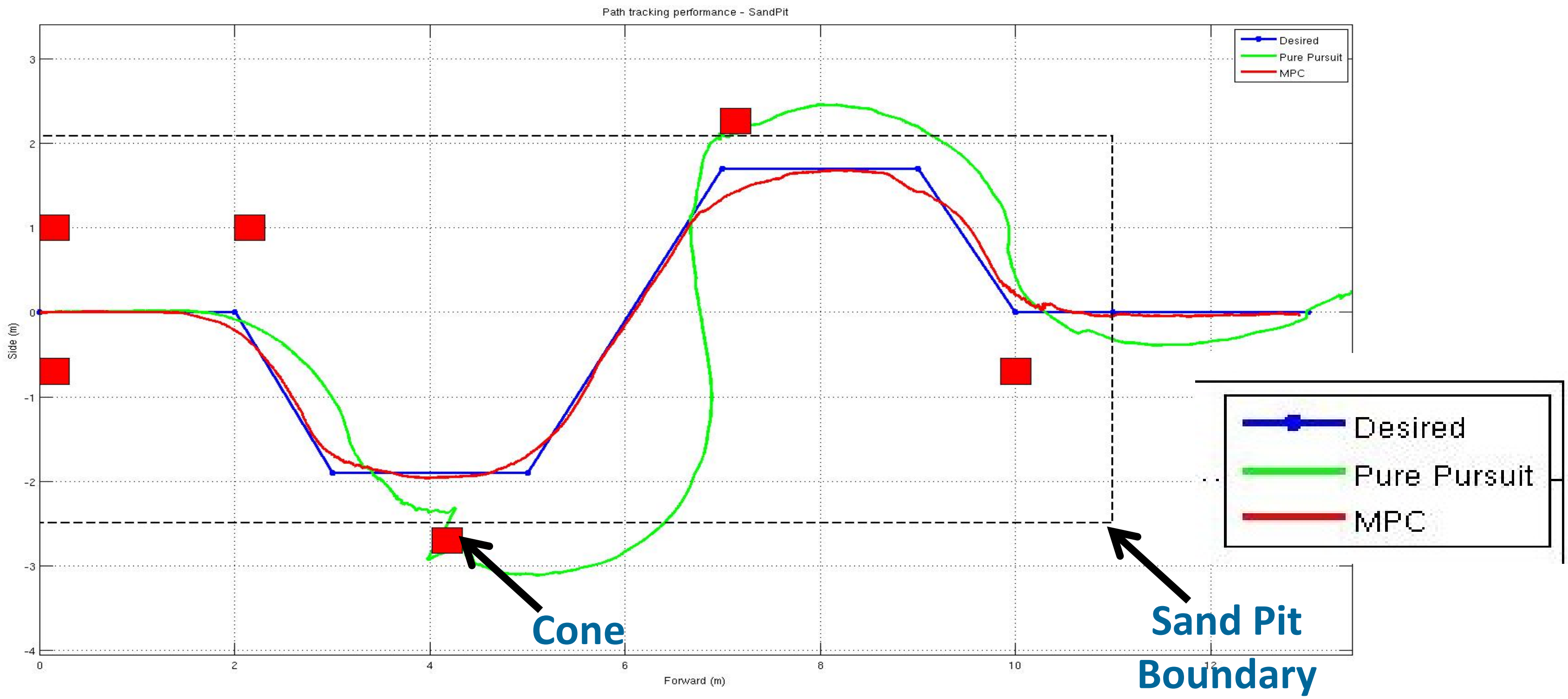


Algorithm/Performance	Pure Pursuit	MPC
Cross Track Error (max/mean) (m)	0.75/0.20	0.3/0.1

MPC Trials – Sand Pit Setup

- The pit was 30 ft long, 15 feet wide and 7 inches deep
- The pit was filled with “fine” river sand – we wanted the vehicle to “slip” and “dig – in” quite a bit as it made the maneuvers
- Any turn >= 90 degrees immobilized the robot
- Trials were conducted on a path that traversed the length of the pit and six 60 degree turns (3 left and 3 right)
- Total Path length was 17.37 meters and max desired speed was 1.0 m/s

MPC Trials – Sand Pit (Performance Plots)



Algorithm/Performance	Pure Pursuit	MPC
Cross Track Error (max/mean) (m)	**1.54/0.17**	**0.26/0.04**

Videos MPC Trials – Sand Pit (MPC)

MPC Conclusion

- Trials in simulation and in the real world show that Predictive Control for accurate path following becomes a <u>necessity</u> as the slip characteristics of the terrain deviates from the ideal conditions of no-slip.

Table Of Contents

- Statement of Problem Studied
- Summary of Most Important Results
- <u>Metrics for Reporting Period</u>
- Conclusion / Outlook

Data for Distribution/Data Paper

- No new papers this time.
- We are planning an IJRR “data paper” to give our data to the community. Such papers are very highly cited.
- Proposed data includes:
  – Moblility logs (post-processed RTK-GPS pose, wheel odometry) for 3 different terrain (grass, dirt, parking lot) on the LandTamer (6x6 skid-steered) and the RecBot (4x2 ackermann) on the same day.
  – Mobility logs from the UPI program on the Crusher (6x6 skid-steered) (may need to get some kind of clearance for releasing pose and odometry data) collected at the following sites – Taylor, Gascola, Somerset, Fort Bliss and Fort Drum.

Table Of Contents

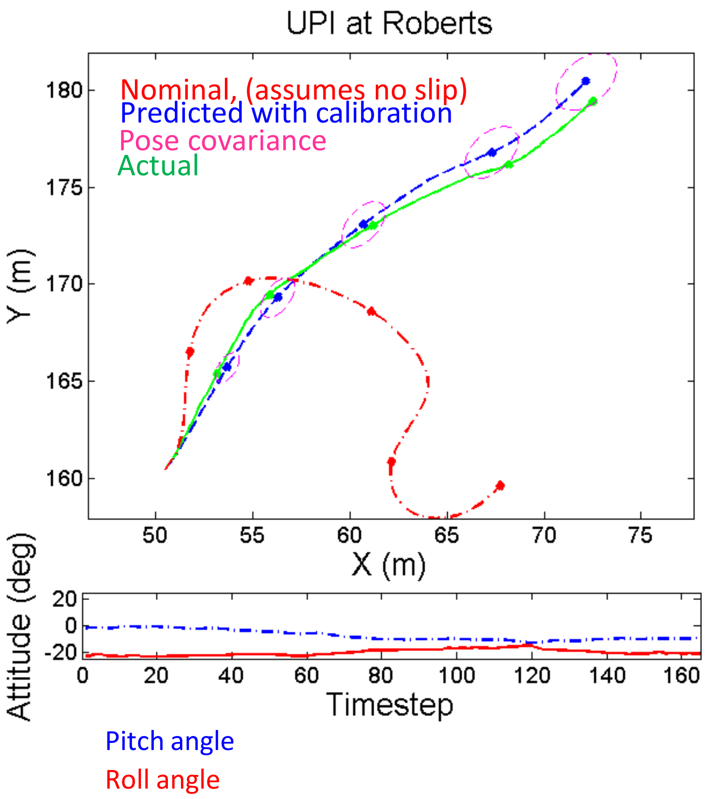
- Statement of Problem Studied
- Summary of Most Important Results
- Metrics for Reporting Period
- Conclusion / Outlook

Program Summary

- Because this is our last report, we want to summarize the most significant conclusions of the program.
- The program addressed four questions (One phase per year):
  - Phase I: Data Gathering, Model Formulation, Off Line Calibration
  - Phase II: Data Gathering and Real-Time Identification
  - Phase III: Incorporation of Perception and Terrain Prediction
  - Phase IV Performance Measurement
- Ultimately, we got a major positive result for each question.

Question 1: Model Formation / Slip Prediction

- **Slip is Predictable. A substantial component of slip behavior is predictable based on measurements of vehicle state of motion or commands. Its uncertainty is also predictable in a statistical sense.**



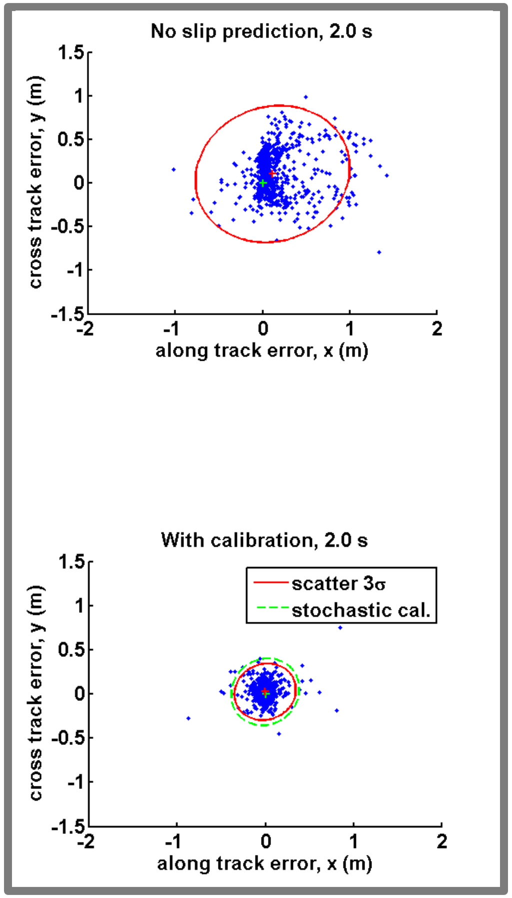
Question 2: On Line Real-Time Identification

- Vehicles can calibrate themselves. Both systematic and stochastic slip models can be identified in real time, in seconds, during system operation on typical trajectories, using sensors already in place.
- We invented a new identification technique to do this and published in IJRR.

$$\underline{\delta x}(t) = \Phi(t,t_0)\underline{\delta x}(t_0) + \int_0^t \Phi(t,\tau)G(\tau)\underline{\delta u}(\tau)\,d\tau$$

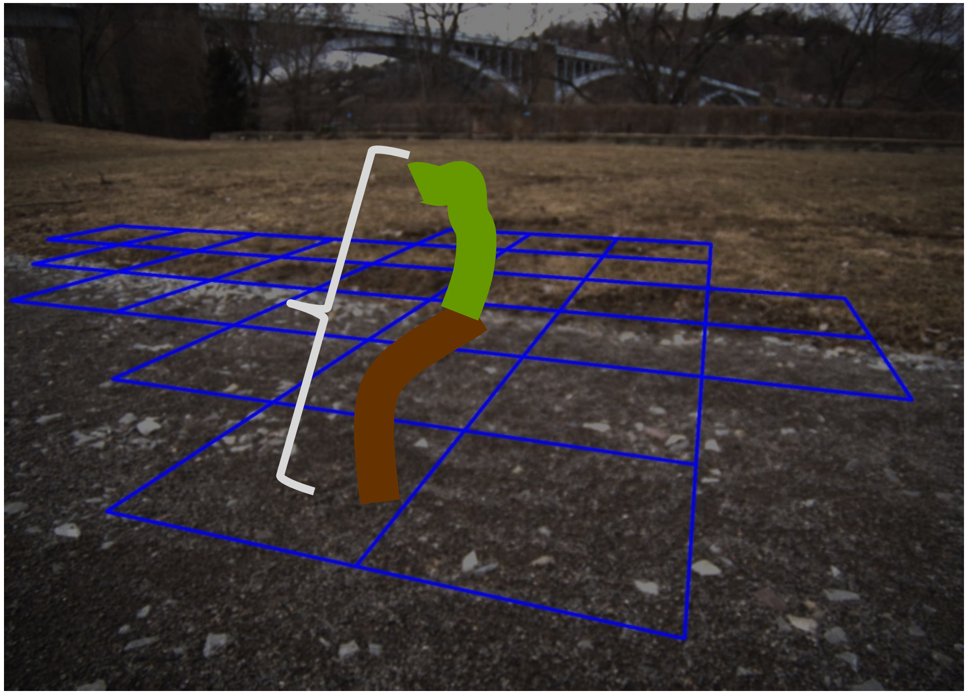
Before

After



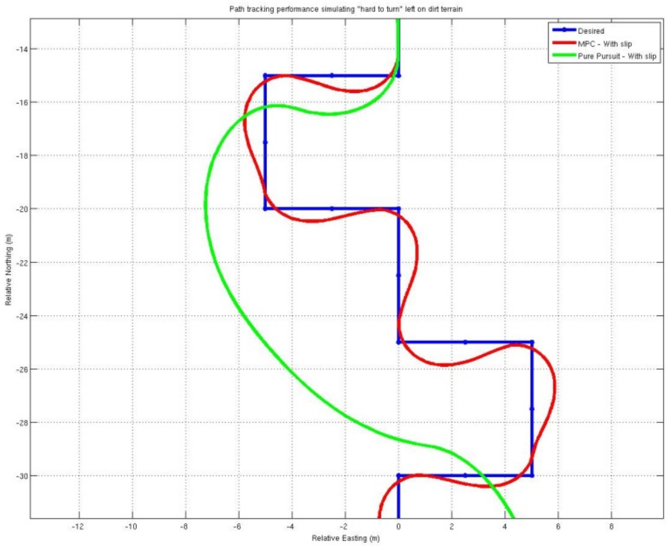
Question 3: Incorporating Perception. Perceptual Cueing

- Perception can Help. Terrain properties observable with EOIR sensing can be correlated with slip experience to learn a mapping from terrain appearance to terrain mechanical properties that affect slip.
- This can be done online, in a totally unsupervised manner, using sensors already on the robot.



Question 4: Predictive Control

- Because we now know it can be done accurately, slip prediction is a major advance in performance in high performance / challenging conditions.
- In our experiments, even a skid steered robot partially buried in the sand can follow a path accurately when a classical controller fails utterly in just a few feet of motion.



Overall Summary

- The implications of our results for future Army robots may be profound.
  - Routine problems become more routine, hard ones become easy, and impossible ones become doable.
- We have shown that robots can learn their own mobility and use the associated models to improve their state estimation and control. This can be done while operating, using the sensors already in place, with no human involvement.
  - Robots can also communicate with others about what they learn about terrain.
- They can make the right decisions about what they can and cannot do, and they can more fully exploit their own performance envelope.

Overall Summary (Military)

- **When we started, we were not sure any of this was possible - or that it would matter. This was high risk, high payoff (6.1) work.**
- **Our work has the following potential impacts on robots (or simply intelligent vehicles of any kind) in the Army. Taken from the proposal, and mostly realized by a few students in 4 years…**
  - … shown the capacity of combined perception, proprioception, prior maps, to advance the performance of UGVs in terms of their capacity to keep up with the force.
  - … advance the state of the art by enabling a new capacity to predict and manage the entrapment hazards which are an important Achilles heel of contemporary UGVs.
  - … enable higher performance UGV planning systems by providing better predictive models. This work should enable more effective obstacle negotiation, high speed path following, and high speed teleoperation.
  - … enable UGVs to compensate, in limited cases, for battle damage by learning the changes in their own mobility models and employing sophisticated controllers to compensate.
  - … enable appropriate behavior changes in UGVs in response to changing terrain and weather conditions. For example, UGVs might alter their plans to avoid certain slopes after a heavy rain.
  - … a step toward enabling rapid adaptation of a fleet of UGVs to changing terrain and weather conditions based on having them share information about their experiences in theater.
  - … enable new simulation technologies like virtualized UGV proving grounds where terrain conditions can be changed by modulating the UGV predictive models.

Overall Summary (General)

- Our work enables or takes us a step closer to all of the following “dreams” for the future.
  - To allow a robot car to deliberately skid sideways into a parking spot.
  - To permit a Mars rover to slide downhill to stop at a science target in precisely the way it intended.
  - To allow an automotive directional and stability control system, or automatic braking system, to sense the terrain and adapt its algorithm to experience.
  - To allow a UGV to automatically extricate itself from an entrapment hazard using something like the way we get our cars out of snow.
  - To allow any robot to execute maneuvers competently through the regime where wheel slip becomes a significant effect on any given terrain.
- We did not show, for lack of time, that self calibration can be turned on and left on forever with no negative consequences. This is important for deployment but it is also expensive to show beyond reasonable doubt.
- We stumbled on a better formulation of the mathematics of wheeled mobile robot dynamics under extreme mobility conditions. It is a particular form of Lagrangian dynamics that is convenient for encoding slip and terrain following etc. and is very very efficient. A grad student has begun exploring this more on the ARL rCTA program. A new ARO program that takes all of our work to the next level could be based on this foundational result.

Acknowledgements

- The following people have contributed in significant ways to the project.
  - Forrest Rogers-Marcovitz, Robotics Engineer, former Masters Student.
  - Neal Seegmiller, Robotics PhD Student
  - Zachary Lamb, Robotics Masters student
  - Venkat Rajagopalan, Robotics Engineer
  - Ammar Husain, Robotics Engineer